CONTENTS

SETTING UP R EDITOR

Exercise: 1

Date: 14.11.2019

Aim:

To know how to setup the R editor

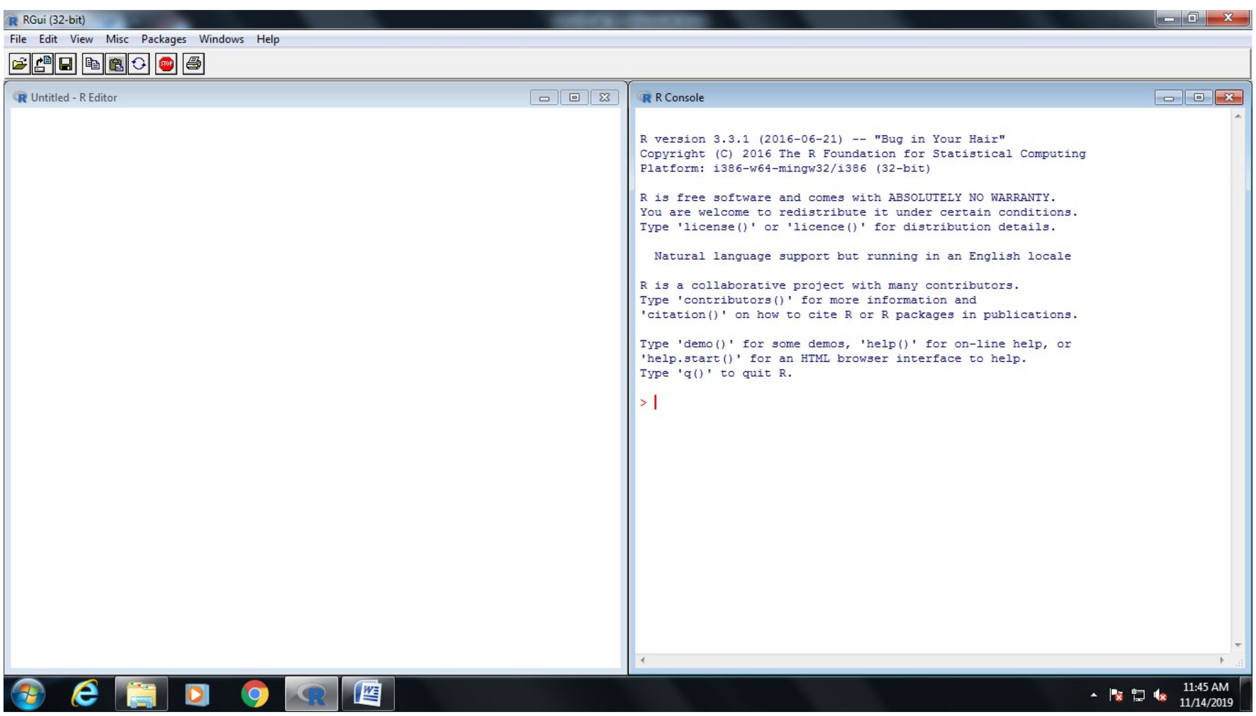
Steps:

Step 1: Open R software

Step 2: Select file ->new script

Step 3: Select windows -> Tile vertically.

R Editor:



Interpretation:

We learned the footsteps to setup the R editor.

MANAGING WORKING DIRECTORY

Exercise: 2

Date: 14.11.2019

Aim:

To know how to manage the current working directory

 # To get the current working Directory

R Code and Output:

```
> getwd()
[1] "C:/Users/ugst46/Desktop"
>
```

#Changing the Current Working Directory

R Code and Output:

```
> setwd("D:")
> getwd()
[1] "D:/"
> setwd("C:/Users/ugst46/Desktop")
> getwd()
[1] "C:/Users/ugst46/Desktop"
>
```

Interpretation:

We learned the footsteps to manage the current working directory

DATA TYPES IN R: VECTOR

Exercise: 3

Date: 14.11.2019

Aim:

To know how to create various types of vectors in R

R Code and Output:

```
> #-----------------------------NUMERIC VECTOR-------------------------#
> age <- c(26,46,45,48,49,34,32,64)
> age
[1] 26 46 45 48 49 34 32 64
> #---------------------------CHARACTER VECTOR-------------------------#
> gender <- c("M","F","M","F","F","M","M","F")
> gender
[1] "M" "F" "M" "F" "F" "M" "M" "F"
> #---------------------------LOGICAL VECTOR---------------------------#
> gender1 <- c(TRUE, FALSE, TRUE, FALSE, FALSE, TRUE, TRUE, FALSE)
> gender1
[1]  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE FALSE
> #---------------------VECTORS USING INBUILT FUNCTION-----------------#
> id <- seq(1,8)
> id1 <- seq(1,16,2)
> id2 <- seq(2,24,3)
> id
[1] 1 2 3 4 5 6 7 8
> id1
[1]  1  3  5  7  9 11 13 15
> id2
[1]  2  5  8 11 14 17 20 23
> values <- rep(10,8)
> values
[1] 10 10 10 10 10 10 10 10
> character <- rep(c("yes", "no"),8)
> character
 [1] "yes" "no"  "yes" "no"  "yes" "no"  "yes" "no"  "yes" "no"  "yes" "no"
[13] "yes" "no"  "yes" "no"

> rand <- runif(8)
> rand
[1] 0.69208971 0.25926773 0.86212513 0.56781413 0.92371753 0.77109469 0.02762834
[8] 0.40553547
> rand1 <- runif(8,20,30)
> rand1
[1] 25.85852 28.46778 26.36216 27.81317 23.69519 28.16213 25.71561 28.88007
> rand2 <- round(runif(8,20,30),0)
> rand2
[1] 22 24 24 26 23 24 28 24

> numb <- numeric(8)
> numb
[1] 0 0 0 0 0 0 0 0
```

Interpretation:

We learned the formation of different types of vectors in R

DATA FRAME

Exercise: 4

Date: 14.11.2019

Aim:

To know how to create a data frame in R and to view a specific element of a dataset.

R Code and Output:

```
> #-------------------------------DATA FRAME ---------------------------#
> patientId <- c(1,2,3,4,5,6,7,8,9,10)
> age <- c(24,46,54,53,46,70,58,64,69,42)
> gender <- c("M","M","F","M","F","F","F","M","M","F")
> diabetetype <- c("Type1", "Type2", "Type2", "Type2", "Type1", "Type1", "Type1", "Type2", "Type2", "Type1")
> status <- c("Poor", "Improved", "Excellent", "Poor", "Improved", "Excellent", "Improved", "Improved", "Poor", "Poor")
> feespaid <- c(200,100,100,200,100,200,100,150,250,150)
> patientdata <- data.frame(patientId,age,gender,diabetetype,status,feespaid)
> fix(patientdata)
> 
```

Data Editor

	patientId	age	gender	diabetetype	status	feespaid	var7
1	1	24	M	Type1	Poor	200	
2	2	46	M	Type2	Improved	100	
3	3	54	F	Type2	Excellent	100	
4	4	53	M	Type2	Poor	200	
5	5	46	F	Type1	Improved	100	
6	6	70	F	Type1	Excellent	200	
7	7	58	F	Type1	Improved	100	
8	8	64	M	Type2	Improved	150	
9	9	69	M	Type2	Poor	250	
10	10	42	F	Type1	Poor	150	
11							
12							
13							
14							
15							
16							
17							
18							
19							

Interpretation:

We learnt the coding how to create a data frame in R and to view a specific element of a dataset.

MATRIX

Exercise: 5

Date: 19.11.2019

R Code and Output:

```
> values<-c(1,2,3,4,5,6,7,8,9)
> A<-matrix(values,nrow=3,ncol=3,byrow=FALSE)
> B<-matrix(values,nrow=3,ncol=3,byrow=TRUE)
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> B
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

Matrix Operation:

R Program:

```
> m1values<-c(7,8,4,6,9,2,4,6,1)
> x<-matrix(m1values,nrow=3,ncol=3,byrow=T)
> m2values<-c(4,6,4,7,8,9,5,1,0)
> y<-matrix(m2values,nrow=3,ncol=3,byrow=T)
> m3values<-c(3,1,0,6,4,8,1,6)
> z<-matrix(m3values,nrow=4,ncol=2,byrow=T)
> x
     [,1] [,2] [,3]
[1,]    7    8    4
[2,]    6    9    2
[3,]    4    6    1
> y
     [,1] [,2] [,3]
[1,]    4    6    4
[2,]    7    8    9
[3,]    5    1    0
```

```
> z
     [,1] [,2]
[1,]    3    1
[2,]    0    6
[3,]    4    8
[4,]    1    6
> |
```

```
> #--------------------TRANSPOSE OF A MATRIX---------------------#
> t(x)
     [,1] [,2] [,3]
[1,]    7    6    4
[2,]    8    9    6
[3,]    4    2    1
> #-------------------- ADDITION OF A MATRIX ---------------------#
> x+y
     [,1] [,2] [,3]
[1,]   11   14    8
[2,]   13   17   11
[3,]    9    7    1
> #-------------------- SUBTRACTION OF A MATRIX ---------------------#
> x-y
     [,1] [,2] [,3]
[1,]    3    2    0
[2,]   -1    1   -7
[3,]   -1    5    1
> #-------------------- MULTIPLICATION OF A MATRIX ---------------------#
> x%*%y
     [,1] [,2] [,3]
[1,]  104  110  100
[2,]   97  110  105
[3,]   63   73   70
```

```
> #-------------------- DETERMINANT OF A MATRIX ---------------------#

> det(x)
[1] -5
```

```
> #-------------------- INVERSE OF A MATRIX ---------------------#
> solve(x)
     [,1] [,2] [,3]
[1,]  0.6 -3.2    4
[2,] -0.4  1.8   -2
[3,]  0.0  2.0   -3
> #-------------------- EIGEN VALUES AND EIGEN VECTORS ---------------------#
> eigen(x)
eigen() decomposition
$values
[1] 16.8037610  0.6523576 -0.4561186

$vectors
           [,1]       [,2]       [,3]
[1,] -0.6715278 -0.8151865 -0.6686285
[2,] -0.6202189  0.5299108  0.2784185
[3,] -0.4054367  0.2338065  0.6895064
```

```
> #-------------------- MATRIX AND ITS FUNCTIONS ---------------------#
> mvalues <- c(7,8,2,4,9,5,6,7,4)
> matA <- matrix(mvalues, nrow=3,ncol=3,byrow=F,dimnames = list(c("X","Y","Z"),c("A","B","C")))
> matA
  A B C
X 7 4 6
Y 8 9 7
Z 2 5 4
> m1values <- c(2,3,4,5,4,3,4,6,5)
> matB <- matrix(m1values, nrow=3,ncol=3,byrow=T,dimnames = list(c("X","Y","Z"),c("A","B","C")))
> matB
  A B C
X 2 3 4
Y 5 4 3
Z 4 6 5
> |
```

```
> matA[1,]  # to get the first row elements of the matrix
A B C
7 4 6
> matA[,3]  # to get the third column elements of the matrix
X Y Z
6 7 4
> matA[2,3] # to get the second row and third column element of the matrix
[1] 7
> matA[1,c(2,3)] # to get the first row and second and third column elements of the matrix
B C
4 6
> matA[-1,] # to get the all elements except the first row elements
  A B C
Y 8 9 7
Z 2 5 4
> |
```

Interpretation:

We learnt the R coding to create matrices and how to perform various operations in matrices.

IMPORTING AND EXPORTING DATA FILES

Exercise: 6

Date: 22.11.2019

Aim:

To know how to Import and Export data files in R.

R Code for Importing Data:

```
> mydata <- read.table("U:/ugst46/Book1.csv", header = T, sep=",")
> mydata
```

R Output for Importing Data:

```
> mydata
   S.No Mean.arterial.blood.pressure Age Weight Body.Surface Heart.Beat
1     1                          105  47   85.4         1.75         63
2     2                          115  49   94.2         2.10         70
3     3                          116  49   95.3         1.98         72
4     4                          117  50   94.7         2.01         73
5     5                          112  51   89.4         1.89         72
6     6                          121  48   99.5         2.25         71
7     7                          121  49   99.8         2.25         69
8     8                          110  47   90.9         1.90         66
9     9                          110  49   89.2         1.83         69
10   10                          114  48   97.7         2.07         64
> 
```

R Code for Exporting Data:

```
> write.table(mtcars,"U:/ugst46/data.csv", sep=",")
> fix(mtcars)
```

R Output for Exporting Data:

	row.names	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
2	Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
3	Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
4	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
5	Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
6	Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1
7	Duster 360	14.3	8	360	245	3.21	3.57	15.84	0	0	3	4
8	Merc 240D	24.4	4	146.7	62	3.69	3.19	20	1	0	4	2
9	Merc 230	22.8	4	140.8	95	3.92	3.15	22.9	1	0	4	2
10	Merc 280	19.2	6	167.6	123	3.92	3.44	18.3	1	0	4	4
11	Merc 280C	17.8	6	167.6	123	3.92	3.44	18.9	1	0	4	4
12	Merc 450SE	16.4	8	275.8	180	3.07	4.07	17.4	0	0	3	3
13	Merc 450SL	17.3	8	275.8	180	3.07	3.73	17.6	0	0	3	3
14	Merc 450SLC	15.2	8	275.8	180	3.07	3.78	18	0	0	3	3
15	Cadillac Fleetwood	10.4	8	472	205	2.93	5.25	17.98	0	0	3	4
16	Lincoln Continental	10.4	8	460	215	3	5.424	17.82	0	0	3	4
17	Chrysler Imperial	14.7	8	440	230	3.23	5.345	17.42	0	0	3	4
18	Fiat 128	32.4	4	78.7	66	4.08	2.2	19.47	1	1	4	1
19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.9	1	1	4	1
21	Toyota Corona	21.5	4	120.1	97	3.7	2.465	20.01	1	0	3	1
22	Dodge Challenger	15.5	8	318	150	2.76	3.52	16.87	0	0	3	2
23	AMC Javelin	15.2	8	304	150	3.15	3.435	17.3	0	0	3	2
24	Camaro Z28	13.3	8	350	245	3.73	3.84	15.41	0	0	3	4
25	Pontiac Firebird	19.2	8	400	175	3.08	3.845	17.05	0	0	3	2
26	Fiat X1-9	27.3	4	79	66	4.08	1.935	18.9	1	1	4	1
27	Porsche 914-2	26	4	120.3	91	4.43	2.14	16.7	0	1	5	2
28	Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.9	1	1	5	2

Interpretation:

We learnt the R coding for Importing and Exporting Data files.

CREATING NEW VARIABLES USING MATHEMATICAL OPERATOR

Exercise: 7

Date: 26.11.2019

Aim:

To know how to create new variables using mathematical operator in R.

R Code and Output for Creating new variable:

```
> patientid <- c(1,2,3,4,5,6,7,8,9,10)
> bmi <- c(23.4,25.5,21,23.6,27.8,20.5,19.5,26.5,28.9,28.6)
> age <- c(25,48,51,29,45,60,23,52,59,23)
> height <- c(142,164,164,177,187,146,153,158,147,149)
> weight <- c(78,77,67,49,79,74,67,64,61,88)
> patientdetails <- data.frame(patientid, bmi, age, height, weight)
> fix(patientdetails)

> #---------------------CREATING NEW VARIABLES USING MATHEMATICAL OPERATOR----------------#
> #---------------------MATHEMATICAL OPERATORS ARE +,-,*,/ -------------------------------#
>
> patientdetails$hwr <- patientdetails$height/patientdetails$weight
> patientdetails$hwr
 [1] 1.820513 2.129870 2.447761 3.612245 2.367089 1.972973 2.283582 2.468750
 [9] 2.409836 1.693182
> patientdetails$age_inmonths <- patientdetails$age*12
> patientdetails$age_inmonths
 [1] 300 576 612 348 540 720 276 624 708 276
> patientdetails1 <- cbind(patientdetails,patientdetails$hwr,patientdetails$age_inmonths)
> patientdetails1

   patientid  bmi age height weight      hwr age_inmonths patientdetails$hwr
1          1 23.4  25    142     78 1.820513          300           1.820513
2          2 25.5  48    164     77 2.129870          576           2.129870
3          3 21.0  51    164     67 2.447761          612           2.447761
4          4 23.6  29    177     49 3.612245          348           3.612245
5          5 27.8  45    187     79 2.367089          540           2.367089
6          6 20.5  60    146     74 1.972973          720           1.972973
7          7 19.5  23    153     67 2.283582          276           2.283582
8          8 26.5  52    158     64 2.468750          624           2.468750
9          9 28.9  59    147     61 2.409836          708           2.409836
10        10 28.6  23    149     88 1.693182          276           1.693182
   patientdetails$age_inmonths
1                          300
2                          576
3                          612
4                          348
5                          540
6                          720
7                          276
8                          624
9                          708
10                         276
```

Data Editor

	patientid	bmi	age	height	weight	var6	var7	var8
1	1	23.4	25	142	78			
2	2	25.5	48	164	77			
3	3	21	51	164	67			
4	4	23.6	29	177	49			
5	5	27.8	45	187	79			
6	6	20.5	60	146	74			
7	7	19.5	23	153	67			
8	8	26.5	52	158	64			
9	9	28.9	59	147	61			
10	10	28.6	23	149	88			
11								
12								
13								
14								
15								
16								
17								
18								
19								

Interpretation:

We learnt the R coding for creating new variables using mathematical operator.

CREATING NEW VARIABLES USING CONDITIONAL STATEMENT

Exercise: 8

Date: 26.11.2019

Aim:

To know how to create new variables using conditional statements If Else and If Else If in R.

R Code and Output:

```
> #-----creating new variable using conditional statement---------------#
> #------------creating new variable using conditional IF ELSE statement -----------#
> patientdetails$status <- ifelse(patientdetails$age >= 58, "Senior Citizen", "Youngsters")
> patientdetails <- cbind(patientdetails, patientdetails$status)
> patientdetails
   patientid  bmi age height weight      hwr age_inmonths         status
1          1 23.4  25    142     78 1.820513          300     Youngsters
2          2 25.5  48    164     77 2.129870          576     Youngsters
3          3 21.0  51    164     67 2.447761          612     Youngsters
4          4 23.6  29    177     49 3.612245          348     Youngsters
5          5 27.8  45    187     79 2.367089          540     Youngsters
6          6 20.5  60    146     74 1.972973          720 Senior Citizen
7          7 19.5  23    153     67 2.283582          276     Youngsters
8          8 26.5  52    158     64 2.468750          624     Youngsters
9          9 28.9  59    147     61 2.409836          708 Senior Citizen
10        10 28.6  23    149     88 1.693182          276     Youngsters
```

	patientid	bmi	age	height	weight	hwr	age_inmonths	status	patientdetails$status	patientdetails$status	bmi_Groups
1	1	23.4	25	142	78	1.820513	300	Youngsters	Youngsters	Youngsters	normal
2	2	25.5	48	164	77	2.12987	576	Youngsters	Youngsters	Youngsters	overweight
3	3	21	51	164	67	2.447761	612	Youngsters	Youngsters	Youngsters	normal
4	4	23.6	29	177	49	3.612245	348	Youngsters	Youngsters	Youngsters	normal
5	5	27.8	45	187	79	2.367089	540	Youngsters	Youngsters	Youngsters	overweight
6	6	20.5	60	146	74	1.972973	720	Senior Citizen	Senior Citizen	Senior Citizen	normal
7	7	19.5	23	153	67	2.283582	276	Youngsters	Youngsters	Youngsters	normal
8	8	26.5	52	158	64	2.46875	624	Youngsters	Youngsters	Youngsters	overweight
9	9	28.9	59	147	61	2.409836	708	Senior Citizen	Senior Citizen	Senior Citizen	overweight
10	10	28.6	23	149	88	1.693182	276	Youngsters	Youngsters	Youngsters	overweight
11											
12											

Interpretation:

To know how to create new variables using conditional statements If Else and If Else If in R

CREATING INDICATOR VARIABLES USING CONDITIONAL STATEMENT

Exercise: 9

Date: 26.11.2019

Aim:

To know how to create Indicator variables using conditional statements

R Code and Output:

```
> #----------------creating Indicator variables--------------------#
> patientdetails$normal <- ifelse(patientdetails$bmi_Groups ==c("Normal"),1,0)
> patientdetails$overweight<-ifelse(patientdetails$bmi_Groups==c("overweight"),1,0)
> patientdetails$Underweight<-ifelse(patientdetails$bmi_Groups==c("Underweight"),1,0)
> fix(patientdetails)
```

	patientid	bmi	age	height	weight	hwr	age_inmonths	status	patientdetails$status	patientdetails$status	bmi_Groups	normal	overweight	Underweight
1	1	23.4	25	142	78	1.820513	300	Youngsters	Youngsters	Youngsters	normal	0	0	0
2	2	25.5	48	164	77	2.12987	576	Youngsters	Youngsters	Youngsters	overweight	0	1	0
3	3	21	51	164	67	2.447761	612	Youngsters	Youngsters	Youngsters	normal	0	0	0
4	4	23.6	29	177	49	3.612245	348	Youngsters	Youngsters	Youngsters	normal	0	0	0
5	5	27.8	45	187	79	2.367089	540	Youngsters	Youngsters	Youngsters	overweight	0	1	0
6	6	20.5	60	146	74	1.972973	720	Senior Citizen	Senior Citizen	Senior Citizen	normal	0	0	0
7	7	19.5	23	153	67	2.283582	276	Youngsters	Youngsters	Youngsters	normal	0	0	0
8	8	26.5	52	158	64	2.46875	624	Youngsters	Youngsters	Youngsters	overweight	0	1	0
9	9	28.9	59	147	61	2.409836	708	Senior Citizen	Senior Citizen	Senior Citizen	overweight	0	1	0
10	10	28.6	23	149	88	1.693182	276	Youngsters	Youngsters	Youngsters	overweight	0	1	0
11														
12														

```
> marks <- read.table (file = "clipboard", sep="\t", header = TRUE)
> marks
   Dept.No X1st.sem X2nd.sem Average
1        1       45       67    56.0
2        2       78       74    76.0
3        3       54       62    58.0
4        4       58       64    61.0
5        5       59       60    59.5
6        6       68       68    68.0
7        7       67       69    68.0
8        8       62       64    63.0
9        9       88       90    89.0
10      10       90       92    91.0
```

Data Editor

	Dept.No	X1st.sem	X2nd.sem	Average	avgclass	var6	var7
1	1	45	67	56	Second class		
2	2	78	74	76	First class		
3	3	54	62	58	Second class		
4	4	58	64	61	First class		
5	5	59	60	59.5	Second class		
6	6	68	68	68	First class		
7	7	67	69	68	First class		
8	8	62	64	63	First class		
9	9	88	90	89	Distinction		
10	10	90	92	91	Distinction		
11							
12							
13							
14							
15							
16							
17							
18							
19							

```
> #----------------------CREATING INDICATOR VARIABLES------------#
> mark$distinction<-ifelse(mark$avgclass==c("Distinction"),1,0)
> mark$firstclass<-ifelse(mark$avgclass==c("First clss"),1,0)
> mark$secondclass<-ifelse(mark$avgclass==c("Second class"),1,0)
> mark
   Dept.No X1st.sem X2nd.sem Average     avgclass distinction firstclass
1        1       45       67    56.0 Second class           0          0
2        2       78       74    76.0  First class           0          0
3        3       54       62    58.0 Second class           0          0
4        4       58       64    61.0  First class           0          0
5        5       59       60    59.5 Second class           0          0
6        6       68       68    68.0  First class           0          0
7        7       67       69    68.0  First class           0          0
8        8       62       64    63.0  First class           0          0
9        9       88       90    89.0  Distinction           1          0
10      10       90       92    91.0  Distinction           1          0
   secondclass
1            1
2            0
3            1
4            0
5            1
6            0
7            0
8            0
9            0
10           0
```

Interpretation:

We learnt the R coding for selecting random samples from the dataset.

SORTING DATASET IN ASCENDING AND DESCENDING ORDER

Exercise: 10

Date: 30.11.2019

Aim:

To know how to sort dataset in ascending and descending order in R

R Code and Output for sorting in Ascending order:

```
> #-----------------------SORTING IN ASCENDING ORDER---------------------#
> mark1<- mark[order(mark$X1st.sem),]
> fix(mark1)
```

Data Editor

	row.names	Dept.No	X1st.sem	X2nd.sem	Average	avgclass	distinction
1	1	1	45	67	56	Second class	0
2	3	3	54	62	58	Second class	0
3	4	4	58	64	61	First class	0
4	5	5	59	60	59.5	Second class	0
5	8	8	62	64	63	First class	0
6	7	7	67	69	68	First class	0
7	6	6	68	68	68	First class	0
8	2	2	78	74	76	First class	0
9	9	9	88	90	89	Distinction	1
10	10	10	90	92	91	Distinction	1
11							
12							
13							
14							
15							
16							
17							
18							
19							

R Code and Output for sorting in Descending order:

```
> #-----------------------SORTING IN DESCENDING ORDER---------------------#
> mark2<- mark[order(-mark$X1st.sem),]
> fix(mark2)
```

	row.names	Dept.No	X1st.sem	X2nd.sem	Average	avgclass	distinction	firstclass	secondclass
1	10	10	90	92	91	Distinction	1	0	0
2	9	9	88	90	89	Distinction	1	0	0
3	2	2	78	74	76	First class	0	0	0
4	6	6	68	68	68	First class	0	0	0
5	7	7	67	69	68	First class	0	0	0
6	8	8	62	64	63	First class	0	0	0
7	5	5	59	60	59.5	Second class	0	0	1
8	4	4	58	64	61	First class	0	0	0
9	3	3	54	62	58	Second class	0	0	1
10	1	1	45	67	56	Second class	0	0	1
11									
12									

\

R Code and Output for sorting dataset with 2 variables:

```
> #----------------------SORTING DATASET WITH RESPECT TO 2 VARIABLES--------------------#
> mark3<-mark[order(mark$X1st.sem,mark$X2nd.sem),]
> fix(mark3)
```

	row.names	Dept.No	X1st.sem	X2nd.sem	Average	avgclass	distinction	firstclass	secondclass
1	1	1	45	67	56	Second class	0	0	1
2	3	3	54	62	58	Second class	0	0	1
3	4	4	58	64	61	First class	0	0	0
4	5	5	59	60	59.5	Second class	0	0	1
5	8	8	62	64	63	First class	0	0	0
6	7	7	67	69	68	First class	0	0	0
7	6	6	68	68	68	First class	0	0	0
8	2	2	78	74	76	First class	0	0	0
9	9	9	88	90	89	Distinction	1	0	0
10	10	10	90	92	91	Distinction	1	0	0
11									
12									

Interpretation:

We learnt the R coding how to sort dataset in ascending and descending order in R

DROP AND KEEP VARIABLES

Exercise: 11

Date: 30.11.2019

Aim:

To know how to drop and keep variables in R

R Code and Output:

```
> #---------------DROP AND KEEP VARIABLES IN DATASET-----------#
> #-----------------DROP VARIABLES IN DATASET------------------#
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Dropping variables:

```
> data<- names(mtcars)%in%c("mpg","cyl")
> new<- mtcars[!data]
> new
                     disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128             78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic          75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla       71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona       120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger    318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9            79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2       120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa         95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L      351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino        145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora       301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E          121.0 109 4.11 2.780 18.60  1  1    4    2
```

Keeping variables:

```
> newl<- mtcars[data]
> newl
                     mpg cyl
Mazda RX4           21.0   6
Mazda RX4 Wag       21.0   6
Datsun 710          22.8   4
Hornet 4 Drive      21.4   6
Hornet Sportabout   18.7   8
Valiant             18.1   6
Duster 360          14.3   8
Merc 240D           24.4   4
Merc 230            22.8   4
Merc 280            19.2   6
Merc 280C           17.8   6
Merc 450SE          16.4   8
Merc 450SL          17.3   8
Merc 450SLC         15.2   8
Cadillac Fleetwood  10.4   8
Lincoln Continental 10.4   8
Chrysler Imperial   14.7   8
Fiat 128            32.4   4
Honda Civic         30.4   4
Toyota Corolla      33.9   4
Toyota Corona       21.5   4
Dodge Challenger    15.5   8
AMC Javelin         15.2   8
Camaro Z28          13.3   8
Pontiac Firebird    19.2   8
Fiat X1-9           27.3   4
Porsche 914-2       26.0   4
Lotus Europa        30.4   4
Ford Pantera L      15.8   8
Ferrari Dino        19.7   6
Maserati Bora       15.0   8
Volvo 142E          21.4   4
```

Interpretation:

We learnt the R coding for dropping and keeping variables in R.

SUBSETTING DATA

Exercise: 12

Date: 3.12.2019

Aim:

To know how to sub-setting data in R.

R Code and Output:

```
> newdata <- mtcars[which(mtcars$cyl==8&mtcars$carb>=4),]
> newdata
                     mpg cyl disp  hp drat    wt  qsec vs am gear carb
Duster 360          14.3   8  360 245 3.21 3.570 15.84  0  0    3    4
Cadillac Fleetwood  10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8  440 230 3.23 5.345 17.42  0  0    3    4
Camaro Z28          13.3   8  350 245 3.73 3.840 15.41  0  0    3    4
Ford Pantera L      15.8   8  351 264 4.22 3.170 14.50  0  1    5    4
Maserati Bora       15.0   8  301 335 3.54 3.570 14.60  0  1    5    8
>
```

```
> #-------------------------subsetting data--------------------------#
> #---------------------Selecting first three rows------------------#
> newdata<- mtcars[1:3,]
> newdata
               mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4     21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710    22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
> newdata <- mtcars[which(mtcars$cyl==8),]
> newdata
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
>
```

Interpretation:

We learnt the R coding for sub setting data in R.

SELECTING RANDOM SAMPLES FROM THE DATASET

Exercise: 13

Date: 3.12.2019

Aim:

To know how to select random sample of size n in R

R Code and Output:

```
> #------------------Select random sample of size n-------------------#
> id<-read.table("U:\\ugst46\\Book1.csv",header=T,sep=",")
> sample1 <- id[sample(1:nrow(id),4,replace=FALSE),]
> sample1
  S.No Mean.arterial.blood.pressure Age Weight Body.Surface Heart.Beat
5    5                          112  51   89.4         1.89         72
8    8                          110  47   90.9         1.90         66
1    1                          105  47   85.4         1.75         63
9    9                          110  49   89.2         1.83         69
>
> 
```

Interpretation:

We learnt the R coding for selecting random samples from the dataset.

AGGREGATE DATASET

Exercise: 14

Date: 3.12.2019

Aim:

To know how to aggregate dataset in R

R Code and Output:

```
> #----------------------aGGREGATE DATASET----------------------------#
> value<- data.frame(custid=c(1,1,2,3,3,3,4,5,5,6,7,7,8,9,9,10),
+ purvalue=c(125,154,136,245,156,398,456,568,459,235,654,547,126,154,187,145))
> value
   custid purvalue
1       1      125
2       1      154
3       2      136
4       3      245
5       3      156
6       3      398
7       4      456
8       5      568
9       5      459
10      6      235
11      7      654
12      7      547
13      8      126
14      9      154
15      9      187
16     10      145
> 
```

```
> valueagg<-aggregate(value$purvalue, by=list(value$custid), FUN = mean)
> valueagg
   Group.1        x
1        1 139.5000
2        2 136.0000
3        3 266.3333
4        4 456.0000
5        5 513.5000
6        6 235.0000
7        7 600.5000
8        8 126.0000
9        9 170.5000
10      10 145.0000
>
> 
```

```
> valueagg<-aggregate(value$purvalue, by=list(value$custid), FUN = sum)
> valueagg
   Group.1    x
1        1  279
2        2  136
3        3  799
4        4  456
5        5 1027
6        6  235
7        7 1201
8        8  126
9        9  341
10      10  145
> valueagg<-aggregate(value$purvalue, by=list(value$custid), FUN = sd)
> valueagg
   Group.1         x
1        1  20.50610
2        2        NA
3        3 122.40234
4        4        NA
5        5  77.07464
6        6        NA
7        7  75.66043
8        8        NA
9        9  23.33452
10      10        NA
> valueagg<-aggregate(value$purvalue, by=list(value$custid), FUN = min)
> valueagg
   Group.1   x
1        1 125
2        2 136
3        3 156
4        4 456
5        5 459
6        6 235
7        7 547
8        8 126
9        9 154
10      10 145
>
> 
```

Interpretation:

We learnt the R coding for aggregating dataset.

MERGING DATASET

Exercise: 15

Date: 4.12.2019

Aim:

To know how to merge dataset in R

R Code and Output:

```
> #---------------------------merging dataset----------------------------#
> customer<- data.frame(custid=c(1,2,3,4,5,6,7,8,9,10),
+ gender=c("M","F","M","F","F","M","F","M","F","M"),
+ age=c(25,35,65,45,28,61,49,54,36,45))
> product<- data.frame(custid=c(1,2,3,4,5,7,8,9,12,13),
+ productcode=c("A1","A2","A3","A4","B1","B2","B3","C1","C2","C3"))
> customer
   custid gender age
1       1      M  25
2       2      F  35
3       3      M  65
4       4      F  45
5       5      F  28
6       6      M  61
7       7      F  49
8       8      M  54
9       9      F  36
10     10      M  45
> product
   custid productcode
1       1          A1
2       2          A2
3       3          A3
4       4          A4
5       5          B1
6       7          B2
7       8          B3
8       9          C1
9      12          C2
10     13          C3
> 
```

```
> #-------------------------FULL JOIN----------------------------#
> mergedata<- merge(customer,product,by.x="custid",by.y="custid",all=T)
> mergedata
   custid gender age productcode
1       1      M  25          A1
2       2      F  35          A2
3       3      M  65          A3
4       4      F  45          A4
5       5      F  28          B1
6       6      M  61        <NA>
7       7      F  49          B2
8       8      M  54          B3
9       9      F  36          C1
10     10      M  45        <NA>
11     12   <NA>  NA          C2
12     13   <NA>  NA          C3
>

> #-------------------------INNER JOIN---------------------------#
> mergedata1<-merge(customer,product,by.x="custid",by.y="custid",all=F)
> mergedata1
  custid gender age productcode
1      1      M  25          A1
2      2      F  35          A2
3      3      M  65          A3
4      4      F  45          A4
5      5      F  28          B1
6      7      F  49          B2
7      8      M  54          B3
8      9      F  36          C1
>
> #-------------------------RIGHT OUTER JOIN----------------------------#
> mergedata2<-merge(customer,product,by.x="custid",by.y="custid",all.y=T)
> mergedata2
   custid gender age productcode
1       1      M  25          A1
2       2      F  35          A2
3       3      M  65          A3
4       4      F  45          A4
5       5      F  28          B1
6       7      F  49          B2
7       8      M  54          B3
8       9      F  36          C1
9      12   <NA>  NA          C2
10     13   <NA>  NA          C3
> |
```

```
> #-------------------------LEFT OUTER JOIN----------------------------#
> mergedata3<-merge(customer,product,by.x="custid",by.y="custid",all.x=T)
> mergedata3
   custid gender age productcode
1       1      M  25          A1
2       2      F  35          A2
3       3      M  65          A3
4       4      F  45          A4
5       5      F  28          B1
6       6      M  61        <NA>
7       7      F  49          B2
8       8      M  54          B3
9       9      F  36          C1
10     10      M  45        <NA>
> 
```

Interpretation:

We learnt the R coding to merge the dataset in R.

STACKING DATSET

Exercise: 16

Date: 5.12.2019

Aim:

To know how to stack dataset in R

R Code and Output:

```
> #------------------------row bind stacking----------------------#
> id<-seq(1,8,1)
> income<-c(18000,15000,21000,16000,17000,16500,14000,13500)
> gender<-c("F","M","M","F","M","F","M","F")
> empdetail<-data.frame(id,income,gender)
> id<-seq(9,16,1)
> income<-c(16000,14000,11000,26000,27000,36500,24000,23500)
> gender<-c("F","M","M","F","M","F","M","F")
> empdetail2<-data.frame(id,income,gender)
> empdata<-rbind(empdetail,empdetail2)
> empdata
   id income gender
1   1  18000      F
2   2  15000      M
3   3  21000      M
4   4  16000      F
5   5  17000      M
6   6  16500      F
7   7  14000      M
8   8  13500      F
9   9  16000      F
10 10  14000      M
11 11  11000      M
12 12  26000      F
13 13  27000      M
14 14  36500      F
15 15  24000      M
16 16  23500      F
>
```

Interpretation:

We learnt the R coding to stack dataset in R.

SIMPLE BARCHART

Exercise: 17

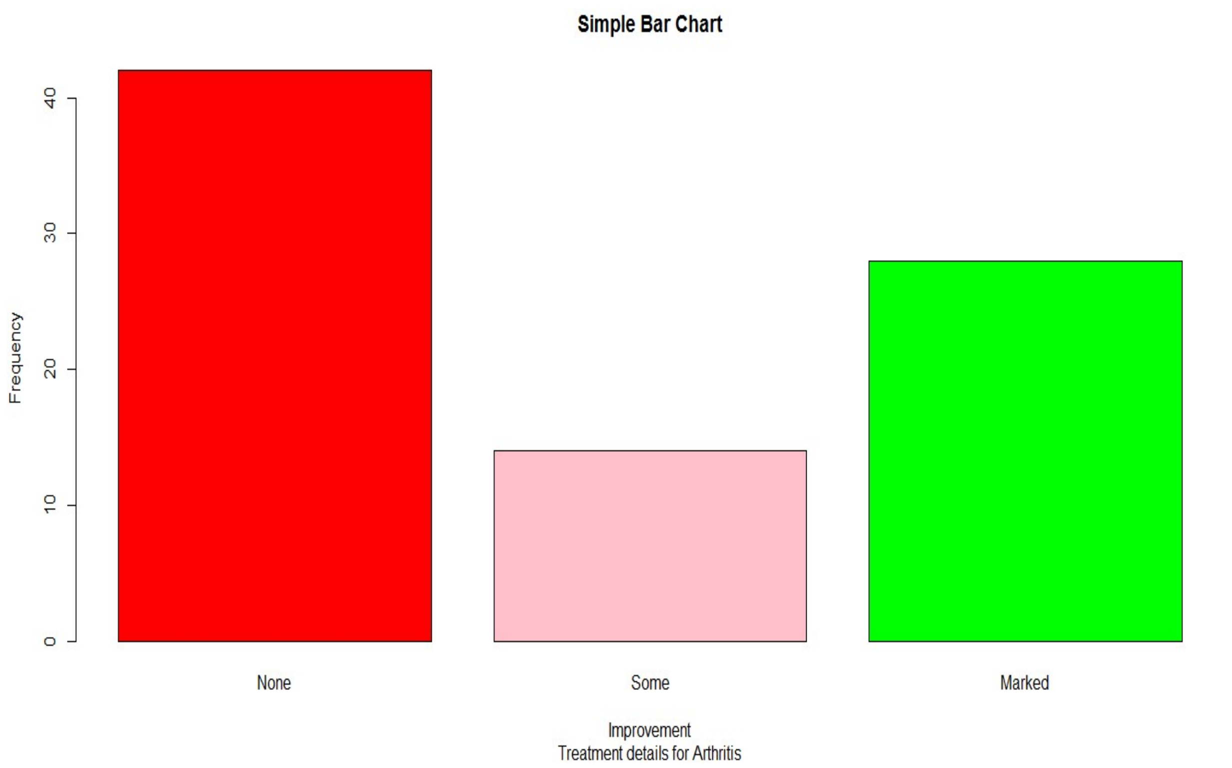
Date: 6.12.2019

Aim:

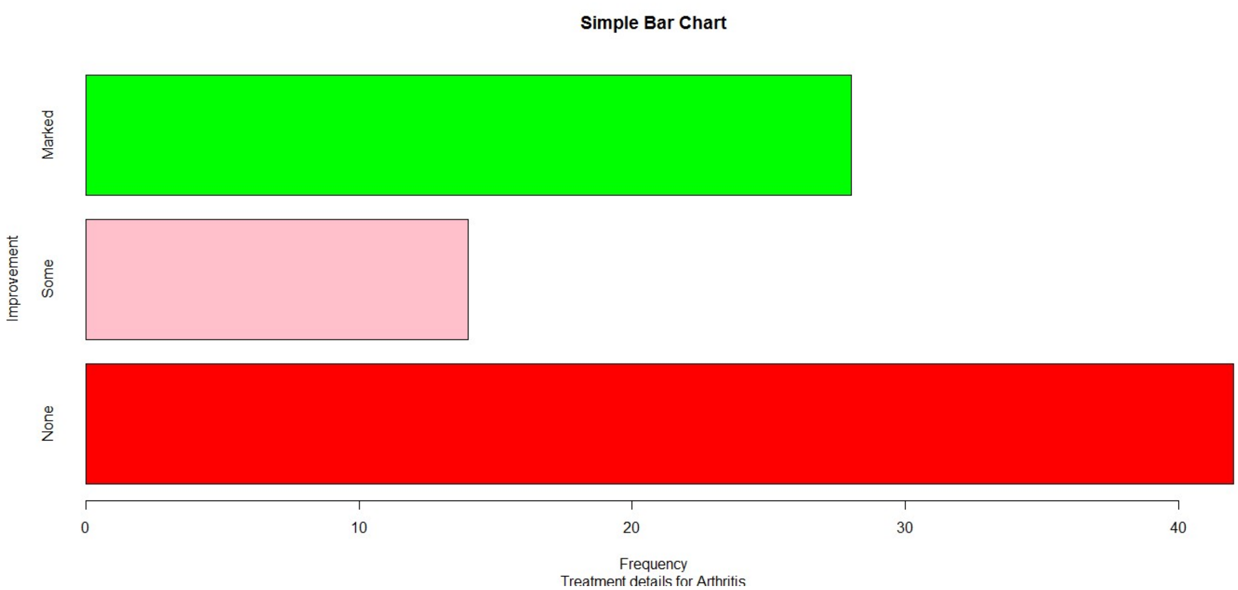
To know how to visualize dataset in R.

R Code and Output:

```
> id<-seq(1,3,1)
> Improvement<-c("None","Some","Marked")
> Frequency<-c(42,14,28)
> #----------SIMPLE BAR CHART USING BOXPLOT----------#
> barplot(Frequency,width=1,main="Simple Bar Chart",sub="Treatment details for
+ Arthritis",names="Improvement",xlab="Improvement",ylab="Frequency",
+ col=c("red","pink","green"), border=T)
```



```
> #----------HORIZONTAL BAR CHART USING BOXPLOT----------#
> barplot(Frequency,width=1,main="Simple Bar Chart",sub="Treatment details for Arthritis",
+ names=Improvement,xlab="Frequency",ylab="Improvement",
+ col=c("red","pink","green"), border=T,horiz=T)
```



Interpretation:

We learnt the R coding to visualize simple bar chart using bar plot function.

LINE CHART

Exercise: 18

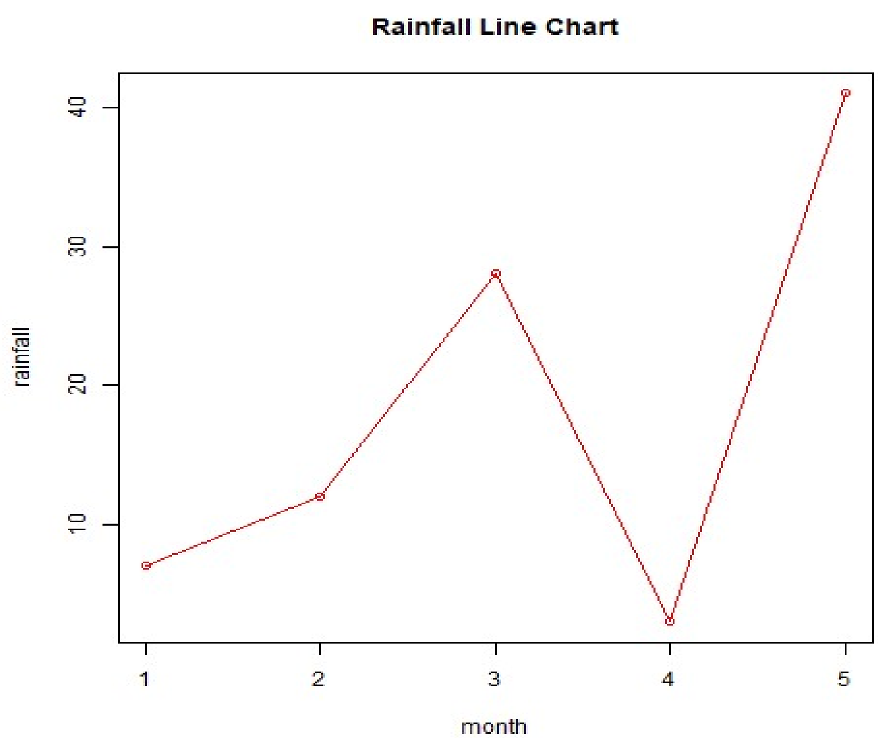
Date: 9.12.2019

Aim:

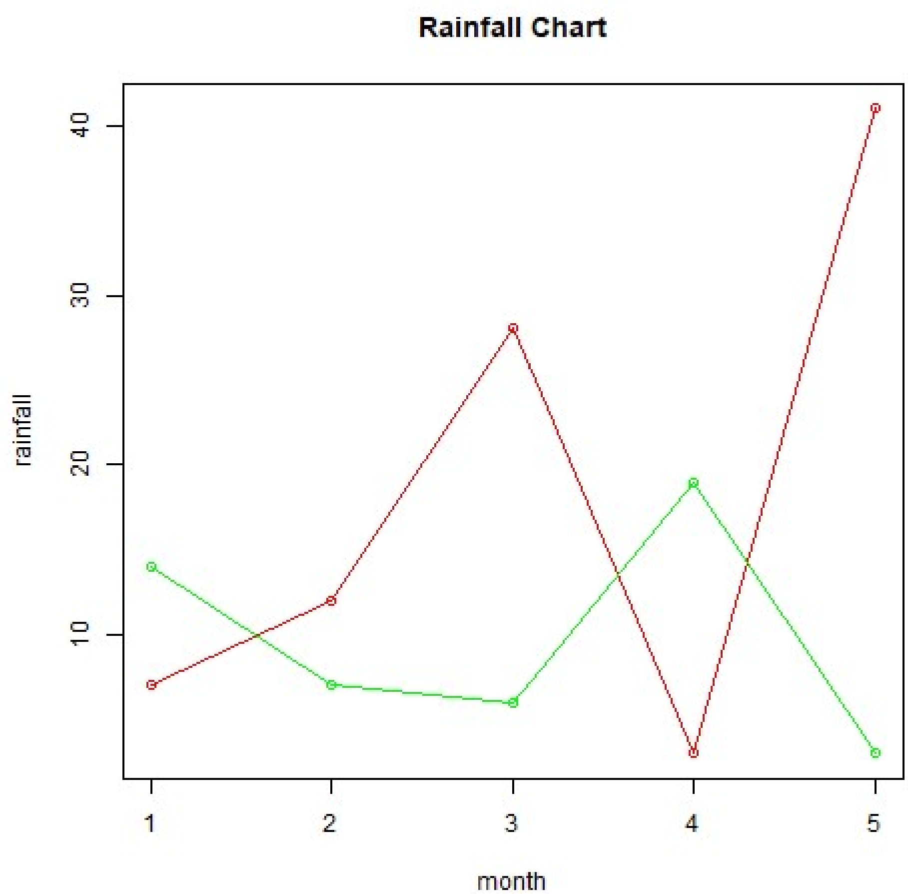
To know how to create line chart in R.

R Code and Output:

```
> #------------------------CREATE THE DATA FOR THE CHART------------------------#
> v<-c(7,12,28,3,41)
> #------------------------GIVE THE CHART FILE NAME-----------------------------#
> png(file="linechart.jpg")
> #-----------------------PLOT THE CHART-------------------------------#
> plot(v,type="o",col="red",xlab="month",ylab="rainfall",main="Rainfall Line Chart"
> #-------------------------SAVE THE FILE------------------------------#
> dev.off()
null device
          1
>
```



```
> #--------------------MULTIPLE LINE IN A CHART-------------------------#
> #----------------------CREATE THE DATA FOR THE CHART-------------------#
> v<-c(7,12,28,3,41)
> t<-c(14,7,6,19,3)
> #---------------------GIVE THE CHART FILE A NAME------------------------#
> png(file="multiplelinechart.jpg")
> #-------------------------PLOT THE CHART-------------------------------#
> plot(v,type="o",col="red",xlab="month",ylab="rainfall",main="Rainfall Chart")
> lines(t,type="o",col="green")
> #--------------------------SAVE THE FILE-------------------------------#
> dev.off()
null device
          1
> 
```



Interpretation:

We learnt the R coding to visualize Line chart in R.

PIE CHART

Exercise: 19

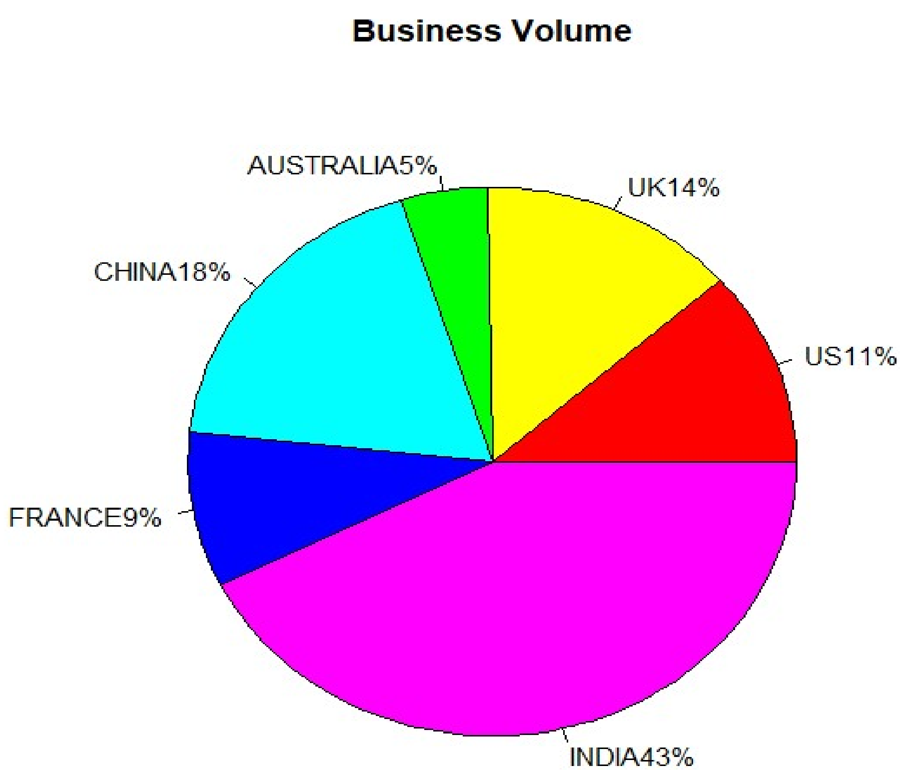
Date: 9.12.2019

Aim:

To know how to visualize Pie charts in R.

R Code and Output:

```
> #--------------------------PIE CHART--------------------------------#
> slices<-c(10,12,4,16,8,37)
> lbls<-c("US","UK","AUSTRALIA","CHINA","FRANCE","INDIA")
> pct<-round(slices/sum(slices)*100)
> lbls2<-paste(lbls,"",pct,"%",sep="")
> pie(slices,labels=lbls2,col=rainbow(length(lbls2)),main="Business Volume")
>
>
> |
```



Interpretation:

We learnt the R coding to visualize Pie Chart in R.

GROUPED BAR CHART

Exercise: 20

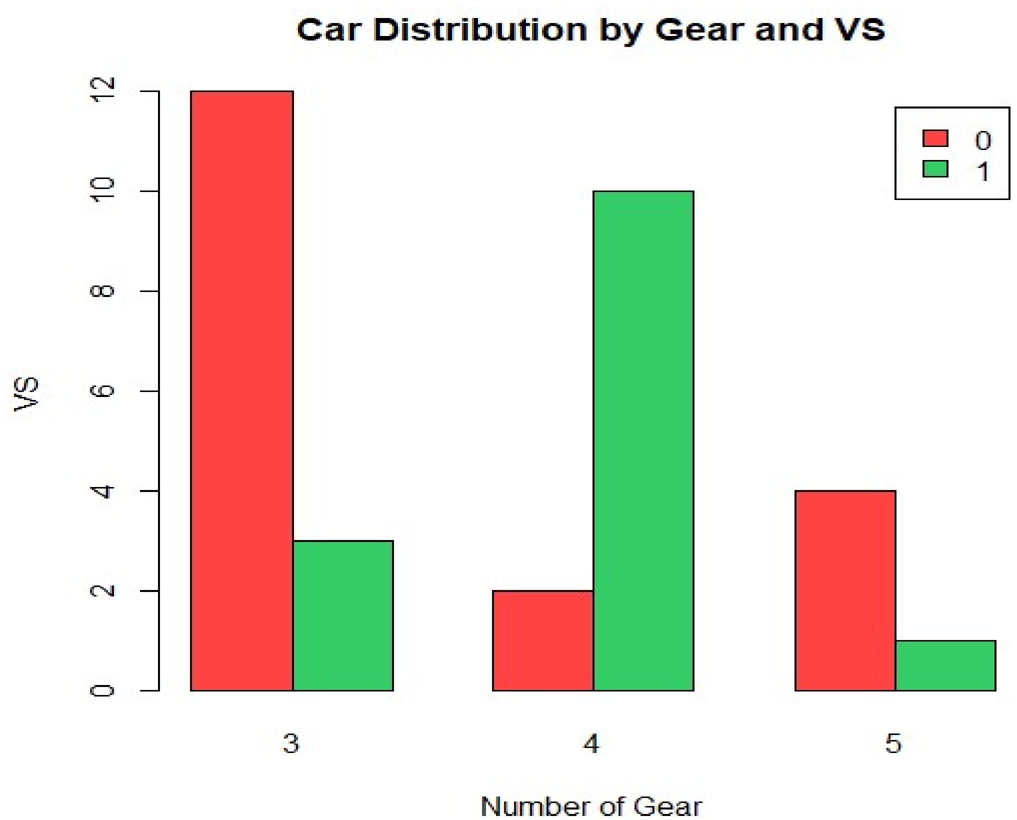
Date: 11.12.2019

Aim:

To know how to visualize (multiple bar diagrams) using bar plotfunction in R.

R Code and Output:

```
> #-----------------------GROUPED BAR CHART---------------------------------#
> counts<-table(mtcars$vs,mtcars$gear)
> barplot(counts,main="Car Distribution by Gear and VS", xlab="Number of Gear",ylab="VS",
+ col=c("#FF4242","#33CC66"),legend=rownames(counts),beside=T)
>
>
> |
```



Interpretation:

We learnt the R coding to visualize (Multiple Bar Diagram) the dataset using bar plot function.

STACKED BAR CHART

Exercise: 21

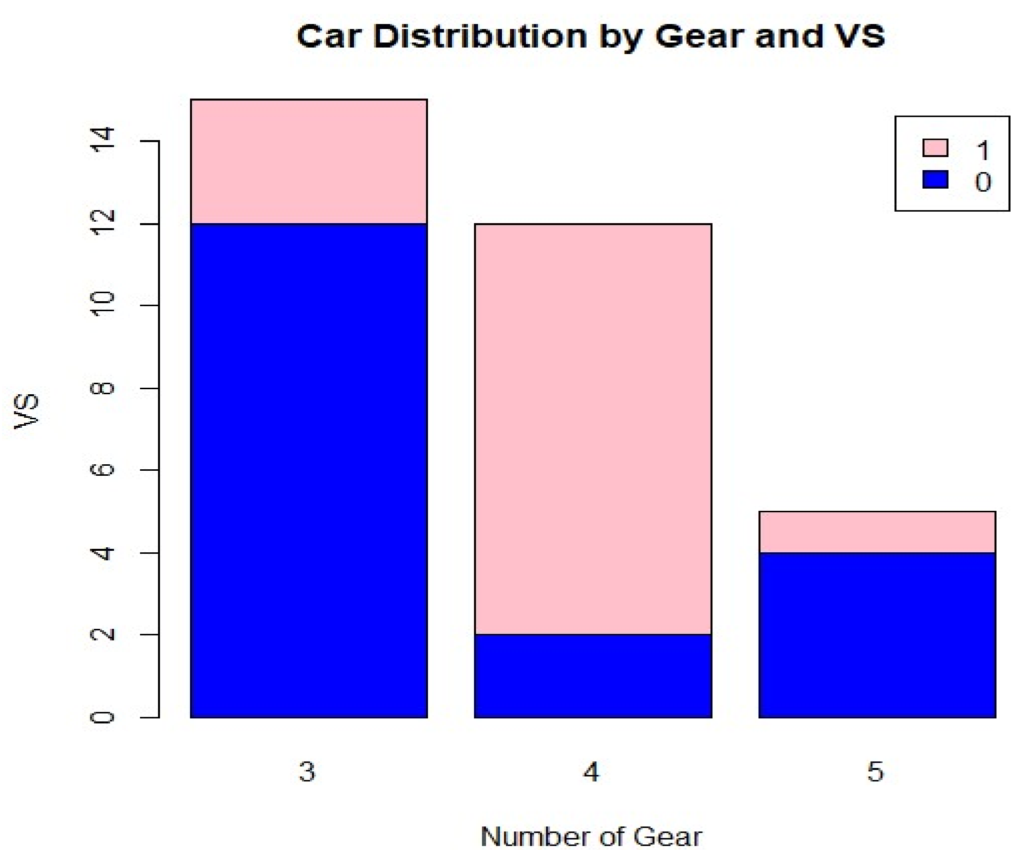
Date: 11.12.2019

Aim:

To know how to visualize stacked Bar Chart the datasets using bar plot function in R.

R Code and Output:

```
> #----------------------STACKED BAR CHART--------------------------------#
> counts<-table(mtcars$vs,mtcars$gear)
> barplot(counts,main="Car Distribution by Gear and VS",xlab="Number of Gear",ylab="VS",
+ col=c("blue","pink"),legend=rownames(counts))
> |
```



Interpretation:

We learnt the R coding to visualize (Stacked Bar Diagram) the dataset using bar plot function.

HISTOGRAM

Exercise: 22

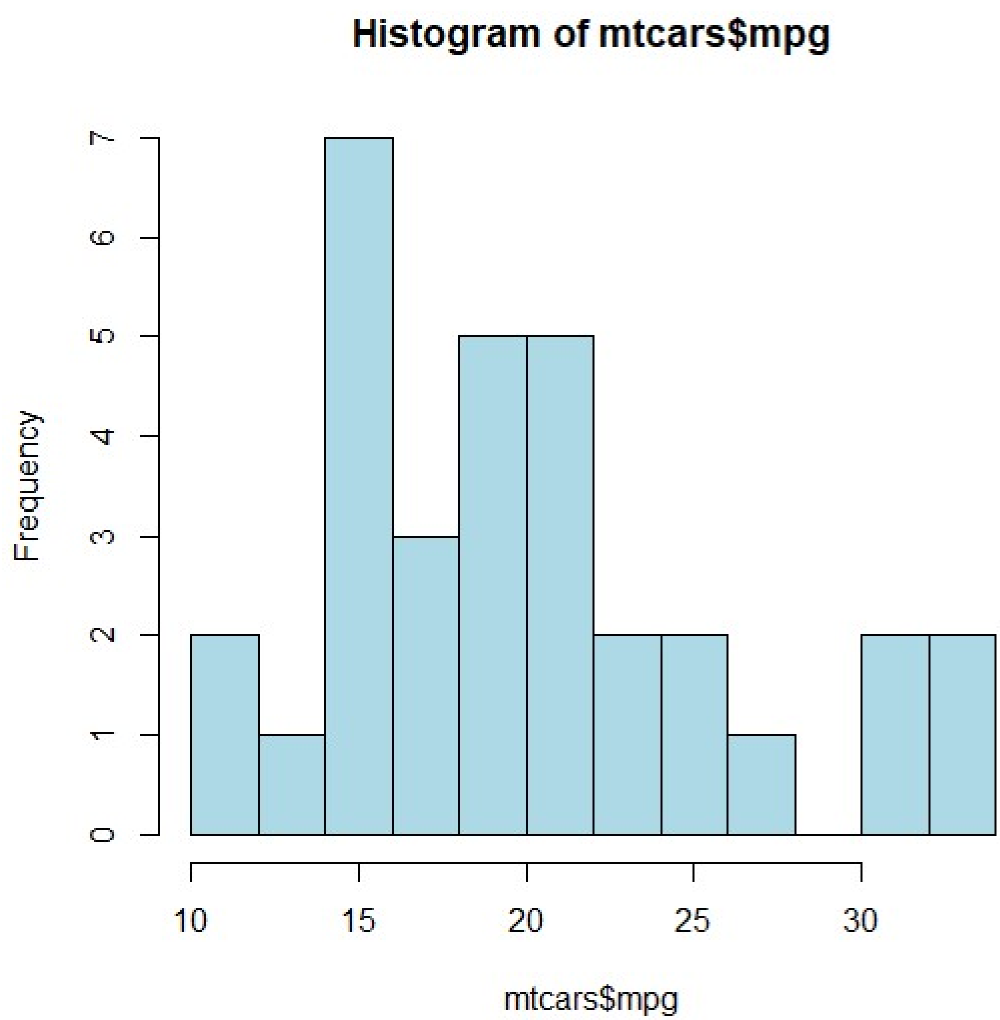
Date: 11.12.2019

Aim:

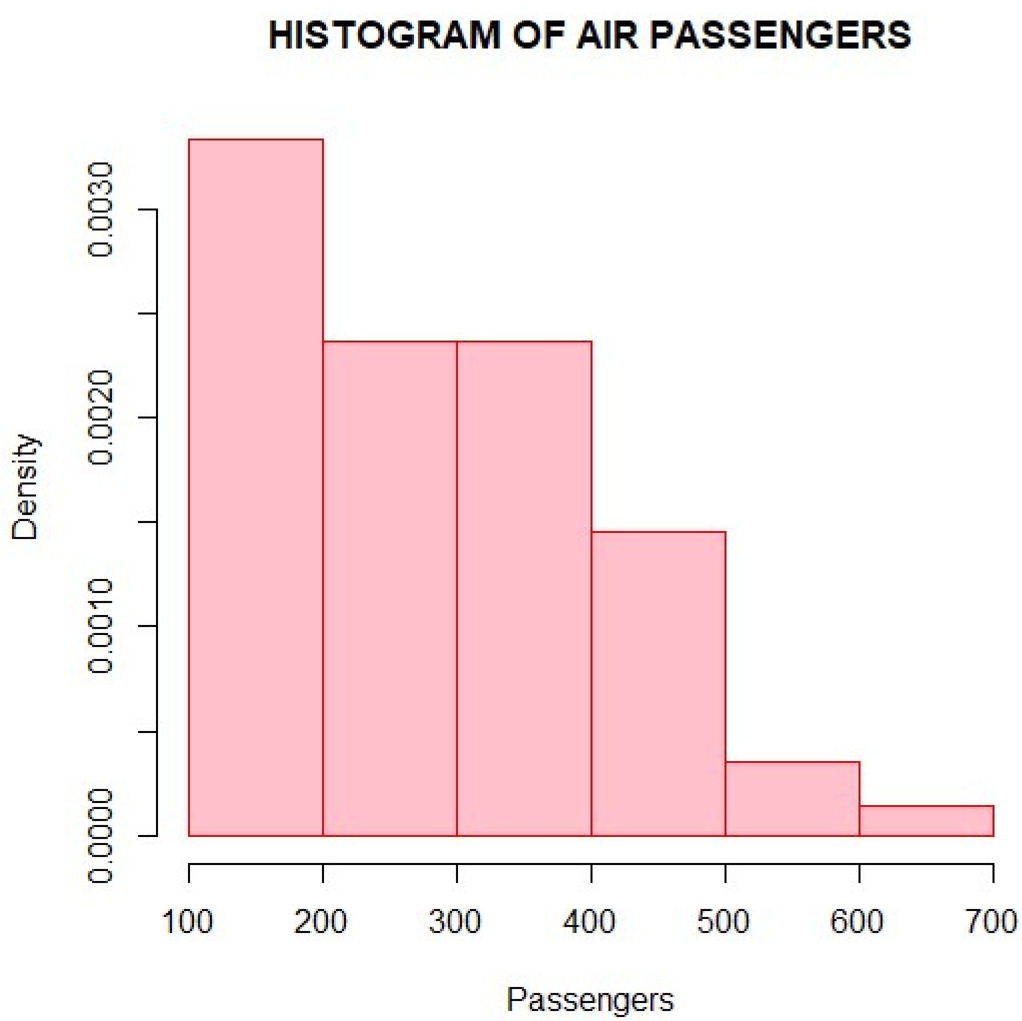
To know how to visualize Histogram in R.

R Code and Output:

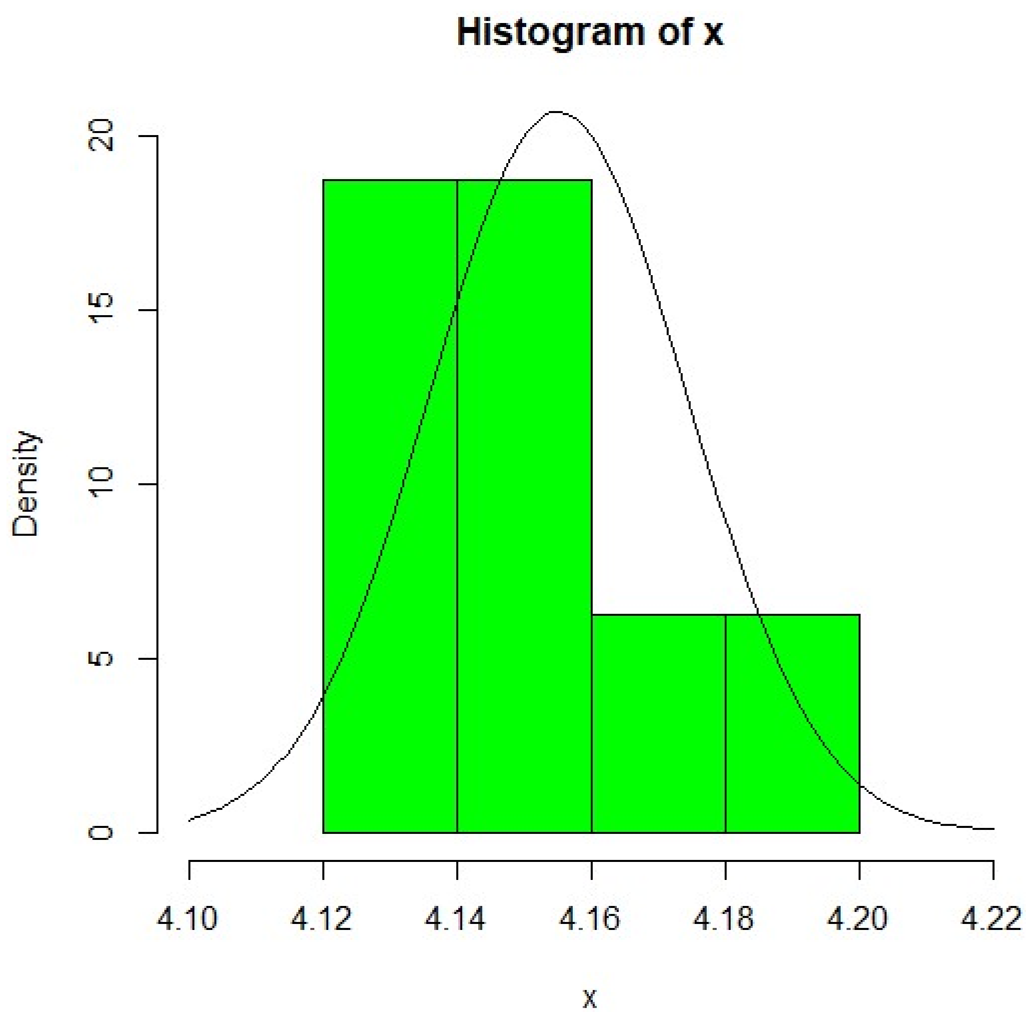
```
> #------------------------HISTOGRAM--------------------------#
> hist(mtcars$mpg,breaks=12,col="light blue")
> 
```



```
> AirPassengers
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
> hist(AirPassengers,main="HISTOGRAM OF AIR PASSENGERS",xlab="Passengers",border="red",col="pink",breaks=5,prob="TRUE")
>
> |
```



```
> #------------------------HISTOGRAM WITH NORMAL CURVE-----------------------#
> x=c(4.14,4.14,4.16,4.15,4.19,4.13,4.16,4.17)
> hist(x)
> hist(x,col="green",xlim=c(4.10,4.22),ylim=c(0,20),probability=TRUE)
> s=sd(x)
> m=mean(x)
> curve(dnorm(x,mean=m,sd=s), add=TRUE)
>
```



Interpretation:

We learnt the R coding to visualize Histogram in R.

BOXPLOT

Exercise: 23

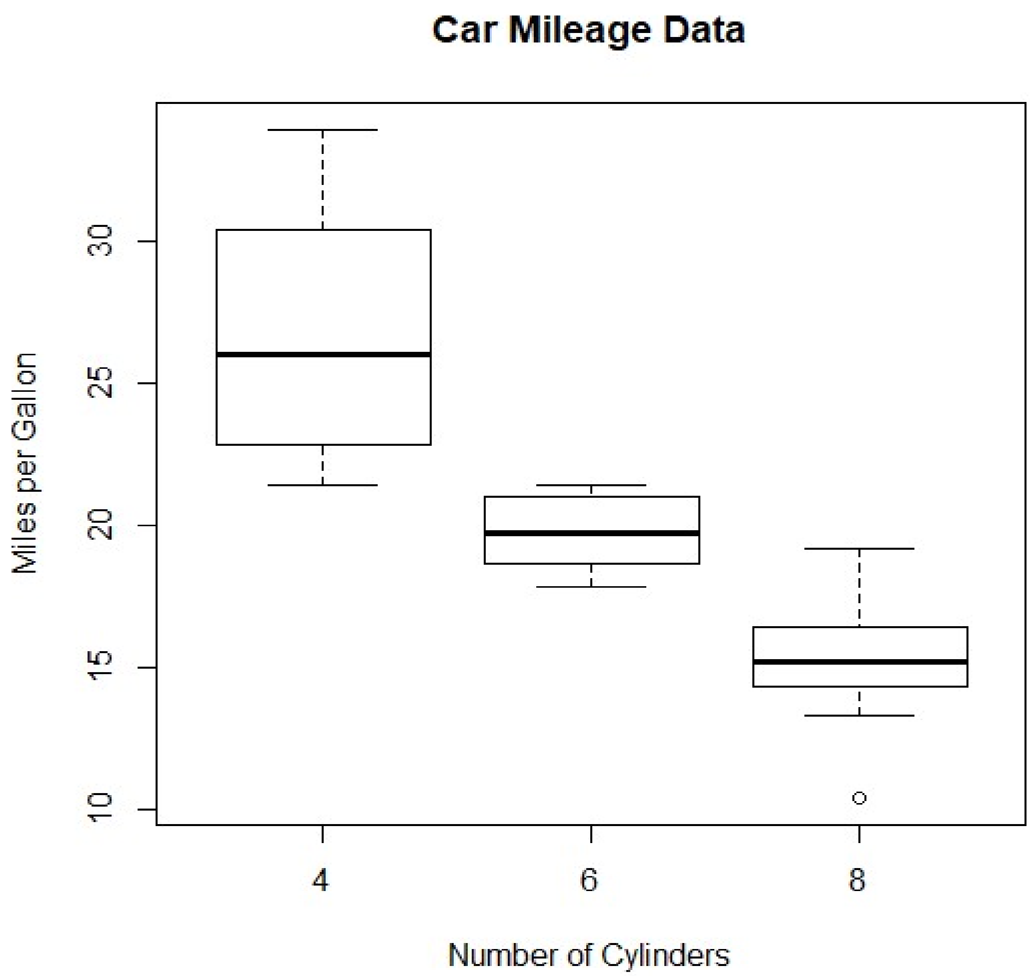
Date: 12.12.2019

Aim:

To know how to visualize Box plot in R.

R Code and Output:

```
> boxplot(mpg~cyl,data=mtcars,main="Car Mileage Data",xlab="Number of Cylinders",ylab="Miles per Gallon")
>
```



Interpretation:

We learnt the R coding to visualize Box plot in R.

SCATTER PLOT

Exercise: 24

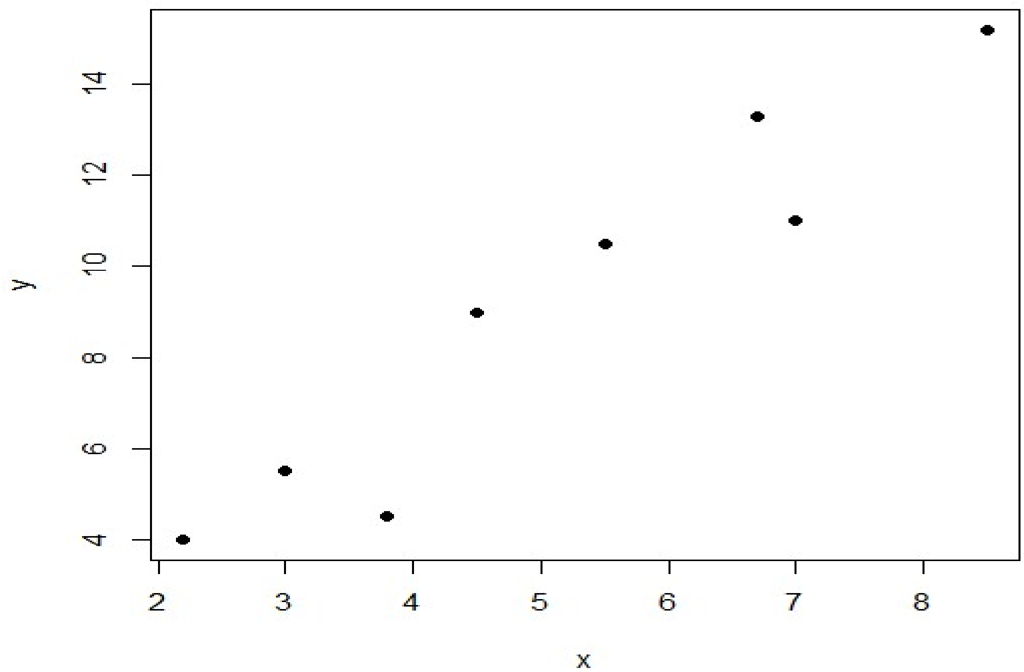
Date: 12.12.2019

Aim:

To know how to visualize Scatter Plot.

R Code and Output:

```
> #-----SCATTER PLOT-----#
> x<-c(2.2,3,3.8,4.5,7,8.5,6.7,5.5)
> y<-c(4,5.5,4.5,9,11,15.2,13.3,10.5)
> #Plot points
> plot(x,y)
> #Changing plottting symbol
> #Use solid circle
> plot(x,y,pch= 19)
```



Interpretation:

We learned the R codings to visualize the Scatter plot in R.

ONE SAMPLE t TEST

Exercise: 25

Date: 31.01.2020

The lifetime of electric bulbs for a random sample of 10 from a large consignment gave the following data:

Item	1	2	3	4	5	6	7	8	9	10
Life in ‘000 hrs	4.2	4.6	3.9	4.1	5.2	3.8	3.9	4.3	4.4	5.6

Can we accept the hypothesis that the average life of bulbs is 4000 hours?

Null Hypothesis:

H_0: The samples have come from the same population.

(Or) The average lifetime of bulbs is 4000 hours.

Alternative Hypothesis:

H_1: The samples have come from the different population.

(Or) The average lifetime of bulbs is not 4000 hours.

R Code and Output:

```
> life<-c(4.2,4.6,3.9,4.1,5.2,3.8,3.9,4.3,4.4,5.6)
> t.test(life,mu=4)

        One Sample t-test

data:  life
t = 2.1483, df = 9, p-value = 0.0602
alternative hypothesis: true mean is not equal to 4
95 percent confidence interval:
 3.978809 4.821191
sample estimates:
mean of x
      4.4
```

Interpretation:

Since the p-value is 0.060 which is greater than 0.05, we accept the null hypothesis. Hence the average lifetime of bulbs is 4000 hours.

ONE SAMPLE t TEST

Exercise: 25

Date: 1.02.2020

The average breaking strength of steel rod is specified to be 17.5. To test a sample of 14 rods was test and the following results were obtained.

15 18 16 21 17 20 19 17 18 17 15 17 20

19

Test at 5% level of significance.

Null Hypothesis:

H_0: The samples have come from the same population.

(Or) The average breaking strength of steel rod is 17.5.

Alternative Hypothesis:

H_1: The samples have come from different population.

(Or) The average breaking strength of steel rod is 17.5.

```
> breakingstrength<-read.table("U:\\ugst46\\2.csv",header = T, sep = ",")
> breakingstrength
   breaking.strength
1                 15
2                 18
3                 16
4                 21
5                 17
6                 20
7                 19
8                 17
9                 18
10                17
11                15
12                17
13                20
14                19
> t.test(breakingstrength,mu=17.5)

        One Sample t-test

data:  breakingstrength
t = 0.57874, df = 13, p-value = 0.5727
alternative hypothesis: true mean is not equal to 17.5
95 percent confidence interval:
 16.71918 18.85225
sample estimates:
mean of x
 17.78571
```

Interpretation:

Since the p value is 0.5727 which is greater than 0.05, we accept the null hypothesis. Hence the average breaking strength of steel rod is 17.5.

T TEST FOR DIFFERENCE OF MEANS (INDEPENDENT SAMPLES)

Exercise: 26

Date: 3.02.2020

To compare the prices of a certain commodity in two town's nine shops were selected at random in each town. The following figures give the price found:

Town A	61	56	63	56	63	59	56	44	61
Town B	55	47	59	51	61	57	54	64	58

Test whether the average price can be said to be the same in two towns.

Null Hypothesis:

H_0: There is no significant difference between the average price of Town A and Town B.

Alternative Hypothesis:

H_1: There is a significant difference between the average price of Town A and Town B.

R Code & output:

```
>
> townA<-c(61,56,63,56,63,59,56,44,61)
> townB<-c(55,47,59,51,61,57,54,64,58)
> t.test(townA,townB,var.equal=TRUE)

        Two Sample t-test

data:  townA and townB
t = 0.55394, df = 16, p-value = 0.5873
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.083341  6.972230
sample estimates:
mean of x mean of y
 57.66667  56.22222
```

Interpretation:

Since the p value is 0.5873 which is greater than 0.05, we accept the null hypothesis. Hence there is no significant difference between the average price of Town A and Town B.

T TEST FOR COMPUTING 2 MEANS WITHOUT ASSUMING EQUAL VARIANCE

Null Hypothesis:

H_0: The true difference in means is equal to zero.

Alternative Hypothesis:

H_1: The true difference in means is not equal to zero.

R Code & output:

```
> t.test(townA,townB)

        Welch Two Sample t-test

data:  townA and townB
t = 0.55394, df = 15.744, p-value = 0.5874
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.090660  6.979549
sample estimates:
mean of x mean of y
 57.66667  56.22222
```

Interpretation:

Since the p value is 0.5874 which is greater than 0.05, we accept the null hypothesis. Hence there is no significant difference between the average price of Town A and Town B.

PAIRED T-TEST (RELATED SAMPLES)

Exercise: 27

Date: 5.02.2020

To verify whether a course in accounting improved performance, a similar test was given to 12 participants both before and after the course.

The original marks recorded in alphabetical order of the participants were,

44 40 61 52 32 44 70 41 67 72 53 and 72.

After the course, the marks were in the same order,

53 38 69 57 46 39 73 48 73 60 and 78.

Was the course useful?

Null Hypothesis:

H_0: There is no significant difference between the average marks of before and after course.

(Or) The course was not useful.

Alternative Hypothesis:

H_1: There is a significant difference between the average marks of before and after course.

(Or) The course was useful (Or) $H_1:\mu_1<\mu_2$.

R Code & output:

```
> before<-c(44,40,61,52,32,44,70,41,67,72,53,72)
> after<-c(53,38,69,57,46,39,73,48,73,74,60,78)
> t.test(before,after,paired=T,alt="less")

        Paired t-test

data:  before and after
t = -3.4454, df = 11, p-value = 0.002736
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
      -Inf -2.393763
sample estimates:
mean of the differences
                     -5
```

Interpretation:

Since the p value is **0.002** which is less than 0.05, we reject the null hypothesis. Hence there is a significant difference between the average marks of before and after course.i.e. The course was useful.

ONE WAY ANOVA

Exercise: 28

Date: 5.02.2020

The following table shows the lives (in hours) of four brands of electric bulbs. Test whether the mean lifetime of bulbs the same across the different brands using one way ANOVA.

Brand	Life of bulbs in Hours							
Philips	1600	1610	1650	1680	1700	1720	1800	
LG	1580	1640	1640	1700	1750			
Surya	1460	1550	1600	1620	1640	1660	1740	1820
Other	1510	1520	1530	1570	1600	1680		

Null Hypothesis:

H_0: There is no significant difference between the average life of bulbs towards brands.

(Or) $H_0:\mu_1=\mu_2=\mu_3=\mu_4$.

Alternative Hypothesis:

H_1:There is no significant difference between the average life of bulbs towards brands.

(Or) $H_1:\mu_1\neq\mu_2\neq\mu_3\neq\mu_4$. (at least one brand differ).

R Code & output:

```
> life<-c(1600,1610,1650,1680,1700,1720,1800,1580,1640,1640,1700,1750,1460,1550,1600,1620,1640,1660,1740,1820,1510,1520,1530,1570,1600,1680)
> brand<-c(rep("Philips",7),rep("LG",5),rep("Surya",8),rep("Other",6))
> time<-data.frame(life,brand)
> results<-aov(life~brand,data=time)
> summary(results)
            Df Sum Sq Mean Sq F value Pr(>F)
brand        3  44361   14787   2.149  0.123
Residuals   22 151351    6880
> |
```

Interpretation:

Since the p value is 0.123 which is greater than 0.05, we accept the null hypothesis. Hence there is no significant difference between the average life of bulbs towards brands.

TWOWAY ANOVA

Exercise: 29

Date: 6.02.2020

Three different methods of analysis M1, M2, M3 are used and the parts per million defective obtained for five different analysts are shown below. Perform a two way analysis of variance and test the significant difference in part per million defective between the three different methods and five different analysts.

		Methods		
		M1	**M2**	**M3**
Analyst	1	7.5	7	7.1
	2	7.4	7.2	6.7
	3	7.3	7	6.9
	4	7.6	7.2	6.8
	5	7.4	7.1	6.9

Null Hypothesis:

H_0: There is no significant difference in part per million defective between the three methods

H_0: There is no significant difference in part per million defective between the five analysts.

Alternative Hypothesis:

H_1: There is a significant difference in part per million defective between the three methods.

H_1: There is a significant difference in part per million defective between the five analysts.

R Code & output:

```
> defective<-c(7.5,7,7.1,7.4,7.2,6.7,7.3,7,6.9,7.6,7.2,6.8,7.4,7.1,6.9)
> Analyst<-c(rep("Analyst 1",3),rep("Analyst 2",3),rep("Analyst 3",3),rep("Analyst 4",3),rep("Analyst 5",3))
> Method<-rep(c("M1","M2","M3"),5)
> ppm<-data.frame(Method,Analyst,defective)
> twoway<-aov(defective~Analyst+Method,data=ppm)
> summary(twoway)
            Df Sum Sq Mean Sq F value   Pr(>F)    
Analyst      4 0.0427  0.0107   0.621 0.660140    
Method       2 0.7960  0.3980  23.184 0.000469 ***
Residuals    8 0.1373  0.0172                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
```

Interpretation:

- Since the p-value for analyst is **0.66** which is greater than 0.05, we accept the null hypothesis.

  There is no significant difference in part per million defective between the five analysts.

- Since the p-value for method is **0.000** which is less than 0.05, we reject the null hypothesis.

  Hence there is a significant difference in part per million defective between the three methods.

LSD (Latin Square Design)

Exercise: 30

Date: 7.02.2020

To analyze the productivity of 5 kind of fertilizer,5 kind of soil and 5 kind of seed. The data are organized in Latin Square design as follows.

	Soil A	Soil B	Soil C	Soil D	Soil E
Fertilizer 1	A42	C47	B55	D51	E44
Fertilizer 2	E45	B54	C52	A44	D50
Fertilizer 3	C41	A46	D57	E47	B48
Fertilizer 4	B56	D52	E49	C50	A43
Fertilizer 5	D47	E49	A45	B54	C46

Null Hypothesis:

H_{01}: There is no significant difference in productivity among different fertilizers.

H_{02}: There is no significant difference in productivity among different soils.

H_{03}: There is no significant difference in productivity among different seeds.

Alternative Hypothesis:

H_{11}: There is a significant difference in productivity among different fertilizers.

H_{12}: There is a significant difference in productivity among different soils.

H_{13}: There is a significant difference in productivity among different seeds.

R Code & output:

```
> values<-c(42,47,55,51,44,45,54,52,44,50,41,46,57,47,48,56,52,49,50,43,47,49,45,54,46)
> fertilizer<-c(rep("fertilizer1",5),rep("fertilizer2",5),rep("fertilizer3",5),rep("fertilizer4",5),rep("fertilizer5",5))
> soil<-c(rep("soilA",1),rep("soilB",1),rep("soilC",1),rep("soilD",1),rep("soilE",1))
> seeds<-c("A","C","B","D","E","E","B","C","A","D","C","A","D","E","B","B","D","E","C","A","D","E","A","B","C")
> Data2<-data.frame(fertilizer,soil,seeds,values)
> Threeway<-aov(values~fertilizer+soil+seeds,Data2)
> summary(Threeway)
            Df Sum Sq Mean Sq F value   Pr(>F)    
fertilizer   4  17.76    4.44   0.797 0.549839    
soil         4 109.36   27.34   4.906 0.014105 *  
seeds        4 286.16   71.54  12.836 0.000271 ***
Residuals   12  66.88    5.57                     
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
```

```
> TukeyHSD(Threeway)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = values ~ fertilizer + soil + seeds, data = Data2)

$fertilizer
                        diff      lwr     upr     p adj
fertilizer2-fertilizer1  1.2 -3.55914 5.95914 0.9245185
fertilizer3-fertilizer1  0.0 -4.75914 4.75914 1.0000000
fertilizer4-fertilizer1  2.2 -2.55914 6.95914 0.5965545
fertilizer5-fertilizer1  0.4 -4.35914 5.15914 0.9986976
fertilizer3-fertilizer2 -1.2 -5.95914 3.55914 0.9245185
fertilizer4-fertilizer2  1.0 -3.75914 5.75914 0.9593153
fertilizer5-fertilizer2 -0.8 -5.55914 3.95914 0.9816941
fertilizer4-fertilizer3  2.2 -2.55914 6.95914 0.5965545
fertilizer5-fertilizer3  0.4 -4.35914 5.15914 0.9986976
fertilizer5-fertilizer4 -1.8 -6.55914 2.95914 0.7485642

$soil
            diff         lwr        upr     p adj
soilB-soilA  3.4  -1.3591399  8.1591399 0.2175142
soilC-soilA  5.4   0.6408601 10.1591399 0.0239813
soilD-soilA  3.0  -1.7591399  7.7591399 0.3181622
soilE-soilA  0.0  -4.7591399  4.7591399 1.0000000
soilC-soilB  2.0  -2.7591399  6.7591399 0.6738074
soilD-soilB -0.4  -5.1591399  4.3591399 0.9986976
soilE-soilB -3.4  -8.1591399  1.3591399 0.2175142
soilD-soilC -2.4  -7.1591399  2.3591399 0.5200518
soilE-soilC -5.4 -10.1591399 -0.6408601 0.0239813
soilE-soilD -3.0  -7.7591399  1.7591399 0.3181622


$seeds
    diff         lwr        upr     p adj
B-A  9.4   4.6408601 14.1591399 0.0003129
C-A  3.2  -1.5591399  7.9591399 0.2642252
D-A  7.4   2.6408601 12.1591399 0.0025012
E-A  2.8  -1.9591399  7.5591399 0.3792691
C-B -6.2 -10.9591399 -1.4408601 0.0095740
D-B -2.0  -6.7591399  2.7591399 0.6738074
E-B -6.6 -11.3591399 -1.8408601 0.0060822
D-C  4.2  -0.5591399  8.9591399 0.0936276
E-C -0.4  -5.1591399  4.3591399 0.9986976
E-D -4.6  -9.3591399  0.1591399 0.0598894
```

Interpretation:

- Since the p-value for fertilizer is **0.54** which is greater than 0.05, we accept the null hypothesis.
  There is no significant difference in productivity among different fertilizers.

- Since the p-value for soil is **0.014** which is lesser than 0.05, we reject the null hypothesis.
  There is a significant difference in productivity among different soils.
  Soil A-C and E-C has a significant difference among the other soils.

- Since the p-value for seeds is **0.0002** which is lesser than 0.05, we reject the null hypothesis.
  There is a significant difference in productivity among different seeds.
  Seeds B-A, D-A, C-B, E-B has a significant difference among the other seeds.

Mann-Whitney U Test

Exercise: 31

Date: 10.02.2020

A large corporate hospital hires most of its doctors from two major universities.

Over the last year, hospital has been conducting test for the newly recruited doctors to determine which university educates better. Based on the following scores, help the human resource department of the hospital to decide whether the universities differ in quality.

Use Mann-Whitney U test.

University A	99	83	89	64	98	85	61	79	91	87	88	
University B	96	90	97	94	86	95	68	78	93	56	76	84

Null Hypothesis:

H_0: There is no significant difference between the scores of University A and University B.

Alternative Hypothesis:

H_1: There is a significant difference between the scores of University A and University B.

R Code & output:

```
> A<-c(99,83,89,64,98,85,61,79,91,87,88)
> B<-c(96,90,97,94,86,95,68,78,93,56,76,84)
> wilcox.test(A,B)

        Wilcoxon rank sum test

data:  A and B
W = 64, p-value = 0.9279
alternative hypothesis: true location shift is not equal to 0
```

Interpretation:

- Since the p-value for fertilizer is **0.9279** which is greater than 0.05, we accept the null hypothesis. There is no significant difference between the scores of University A and University B.

WILCOXON SIGNED RANK TEST

Exercise: 32

Date: 12.02.2020

Performance scores of 16 salesmen before and after training are given below:

Scores Before Training	85	76	64	59	72	68	43	54	57	61	71	82	39	51	54	57
Scores after training	82	79	68	52	75	69	40	53	50	67	71	83	54	59	51	58

At 5% level of significance, test the hypothesis, using wilcoxon test, that the training has not caused any change in the performance score.

Null Hypothesis:

H_0: There is no significant difference between the scores of before and after training.

Alternative Hypothesis:

H_1: There is a significant difference between the scores of before and after training.

R code and output:

```
> before<-c(85,76,64,59,72,68,43,54,57,61,71,82,39,51,54,57)
> after<-c(82,79,68,52,75,69,40,53,50,67,71,83,54,59,51,58)
> wilcox.test(before,after,paired=T,alt="less")

        Wilcoxon signed rank test with continuity correction

data:  before and after
V = 48.5, p-value = 0.2648
alternative hypothesis: true location shift is less than 0
```

Interpretation:

Since the p value is **0.2648** which is less than 0.05, we reject the null hypothesis. Hence there is no significant difference between the scores of before and after training.

KRUSKAL WALLIS TEST

Exercise: 33

Date: 12.02.2020

A departmental store has shops at three different locations in the city. The owner keeps a daily record for each location of the number of customers who actually make a purchase. A sample of those data as follows. Using Kruskal Wallis test, can you say at 5% level of significant the shops have the same number of customer who buy?

Location A	99	64	101	85	79	88	97	95	90	100
Location B	83	102	125	61	91	96	94	89	93	75
Location C	89	98	56	105	87	90	87	101	76	89

Null Hypothesis:

H_0: There is no significant difference between the three locations.

Alternative Hypothesis:

H_1: There is a significant difference between the three locations.

R code and output:

```
> locationA<-c(99,64,101,85,79,88,97,95,90,100)
> locationB<-c(83,102,125,61,91,96,94,89,93,75)
> locationC<-c(89,98,56,105,87,90,87,101,76,89)
> kruskal.test(list(locationA,locationB,locationC))

        Kruskal-Wallis rank sum test

data:  list(locationA, locationB, locationC)
Kruskal-Wallis chi-squared = 0.19643, df = 2, p-value = 0.9065
```

Interpretation:

Since the p value is **0.9065** which is greater than 0.05, we accept the null hypothesis. Hence there is no significant difference between the three locations.

CORRELATION

Exercise: 34

Date: 13.02.2020

AIM:

To create R coding to find Pearson correlation and their testing.

Null Hypothesis:

H_0: There is no correlation between miles per gallon and horse power.

Alternative Hypothesis:

H_1: There is a correlation between miles per gallon and horse power.

R code and output:

```
> cor(mtcars$mpg,mtcars$hp,method="pearson")
[1] -0.7761684
>
> cor.test(mtcars$mpg,mtcars$hp,method="pearson")

        Pearson's product-moment correlation

data:  mtcars$mpg and mtcars$hp
t = -6.7424, df = 30, p-value = 1.788e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8852686 -0.5860994
sample estimates:
       cor
-0.7761684
```

Interpretation:

We learnt the R coding for Pearson correlation correlation.

The Pearson correlation coefficient between miles per gallon and horse power is **-0.7761684** (High negative Correlation). This shows that the car with high horse-power produces less mpg. The p value for testing correlation ($\rho = 0$) is **0.0000001788** is less than 0.01 which indicates that there is a highly significant negative correlation between horse power and miles per gallon.

SPEARMAN AND KENDALL CORRELATION

R code and output:

Candidate	Interviewer 1	Interviewer 2
A	1	1
B	2	2
C	3	4
D	4	3
E	5	6
F	6	5
G	7	8
H	8	7
I	9	10
J	10	9
K	11	12
L	12	11

```
> int1<-c(1,2,3,4,5,6,7,8,9,10,11,12)
> int2<-c(1,2,4,3,6,5,8,7,10,9,12,11)
> cor(int1,int2,method="spearman")
[1] 0.965035
>
```

Interpretation:

Since the p-value is 0.96 which is greater than 0.05 we accept the null hypothesis. There is no correlation between interviewer 1 and interviewer 2.

FITTING OF SIMPLE LINEAR REGRESSION MODEL

Exercise: 35

Date: 14.02.2020

Problem:

A sales Manager collected the following data on annual sales and years of experience.

S.NO	1	2	3	4	5	6	7	8	9	10
Years of experience	1	3	4	4	6	8	10	10	11	13
Annual Sales (1000$)	80	97	92	102	102	111	119	123	117	136

Aim: To fit a Linear Regression Model for the given data

Null Hypothesis:

H_0: The fitted model is not significant.

Alternative Hypothesis:

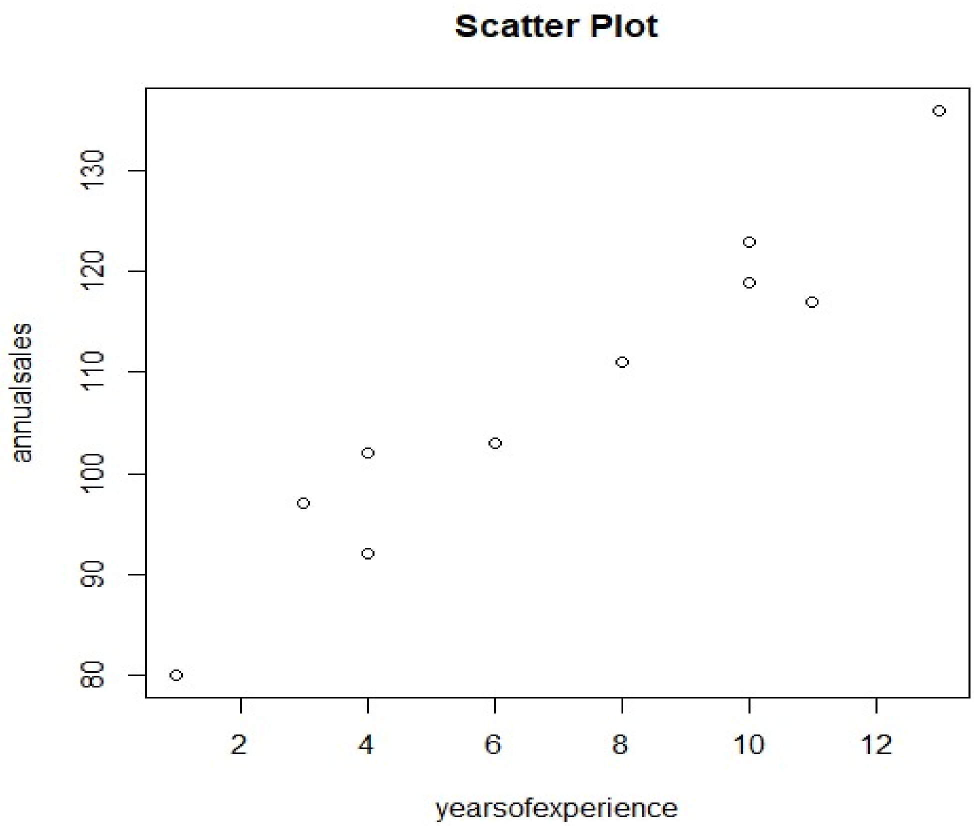
H_1: The fitted model is significant.

R Code and Output:

```
> #-------FITTING OF SIMPLE LINEAR REGRESSION MODEL-------------#
> #---------------GETTING INPUT DATA-------------------#
> salesperson=c(1:10)
> yearsofexperience=c(1,3,4,4,6,8,10,10,11,13)
> annualsales=c(80,97,92,102,103,111,119,123,117,136)
> #-------------GETTING DATA OUTPUT-------------#
> salesdetails=data.frame(salesperson,yearsofexperience,annualsales)
> salesdetails
   salesperson yearsofexperience annualsales
1            1                 1          80
2            2                 3          97
3            3                 4          92
4            4                 4         102
5            5                 6         103
6            6                 8         111
7            7                10         119
8            8                10         123
9            9                11         117
10          10                13         136
```

SCATTER PLOT:

```
> plot(yearsofexperience,annualsales,main="Scatter Plot")
```



CORRELATION:

```
> cor.test(yearsofexperience,annualsales)

        Pearson's product-moment correlation

data:  yearsofexperience and annualsales
t = 10.34, df = 8, p-value = 6.609e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8529446 0.9918346
sample estimates:
      cor 
0.9645646 
```

```
> #----------REGRESSION FITTING------------#
> regmodel=lm(annualsales~yearsofexperience)
> summary(regmodel)

Call:
lm(formula = annualsales ~ yearsofexperience)

Residuals:
   Min     1Q Median     3Q    Max 
 -7.00  -3.25  -1.00   3.75   6.00 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)        80.0000     3.0753   26.01 5.12e-09 ***
yearsofexperience   4.0000     0.3868   10.34 6.61e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.61 on 8 degrees of freedom
Multiple R-squared:  0.9304,	Adjusted R-squared:  0.9217 
F-statistic: 106.9 on 1 and 8 DF,  p-value: 6.609e-06
```

Interpretation:

We learned the R coding for fitting simple linear regression model.

1. From the scatter plot, we can conclude that annual sales and year of experience has linear relationship.
2. The best fitted simple linear regression model is **Y=80+4X.**
3. **92.17%** variations in Y are explained by the variable X.
4. There are $80,000 annual sales without years of experience.
5. Since p-value <0.05 (0.000006609<0.05) at 5% level we reject H_0 and conclude that the fitted model is significant.

FITTING OF MULTIPLE LINEAR REGRESSION MODEL

Exercise: 36

Date: 17.02.2020

The owner of show time movie theatre would like to estimate weekly gross revenue as a function of advertising expenditures. Historical data for sample of eight week as follows:

Weekly Gross Revenue($1000)	96	90	95	92	95	94	94	94
TV advertisement($1000)	5.0	2.0	4.0	2.5	3.0	3.5	2.5	3.0
News paper advertisement($1000)	1.5	2.0	1.5	2.5	3.3	2.3	4.2	2.5

Aim:

To fit a multiple linear regression model for the given data.

Null Hypothesis:

H_0: The fitted model is not significant.

Alternative Hypothesis:

H_1: The fitted model is significant.

R Code and Output:

```
> y<-c(96,90,95,92,95,94,94,94)
> x1<-c(5,2,4,2.5,3,3.5,2.5,3)
> x2<-c(1.5,2,1.5,2.5,3.3,2.3,4.2,2.5)
> data<-data.frame(y,x1,x2)
> reg=lm(y~x1+x2)
> summary(reg)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
      1       2       3       4       5       6       7       8 
-0.6325 -0.4124  0.6577 -0.2080  0.6061 -0.2380 -0.4197  0.6469 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  83.2301     1.5739  52.882 4.57e-08 ***
x1            2.2902     0.3041   7.532 0.000653 ***
x2            1.3010     0.3207   4.057 0.009761 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6426 on 5 degrees of freedom
Multiple R-squared:  0.919,	Adjusted R-squared:  0.8866 
F-statistic: 28.38 on 2 and 5 DF,  p-value: 0.001865
```

Interpretation:

We learnt the R coding for fitting Multiple Linear regression Model

1. The best fitted Multiple linear regression model is
   Y=83.2301+2.2902x1+1.3010x2
2. **88%** variations in Y are explained by the variables x1 & x2.
3. There are **$83230.1** sales without x1 and x2.
4. Since p value < 0.05(0.001865<0.05) at 5% level we reject H_0 and conclude that the fitted model is significant.

SIMPLEX METHOD

Exercise: 37

Date: 19.02.2020

Solve the following LPP by using Simplex method.

$$\text{Max } Z = 22x_1 + 30x_2 + 25x_3$$

Subject to the constraints $2x_1 + 2x_2 + 0x_3 \leq 100$

$$2x_1 + x_2 + x_3 \leq 100$$

$$x_1 + 2x_2 + 2x_3 \leq 100 \; ; \; x_1, x_2, x_3 \geq 0$$

Aim: To create R coding to solve the LPP by simplex method.

R code and Output:

```
> library(boot)
> #----------------SIMPLEX METHOD--------------#
> objective<-c(22,30,25)
> cons1<-c(2,2,0)
> cons2<-c(2,1,1)
> cons3<-c(1,2,2)
> simplex(a=objective,A1=rbind(cons1,cons2,cons3),b1=c(100,100,100),maxi=T)

Linear Programming Results

Call : simplex(a = objective, A1 = rbind(cons1, cons2, cons3), b1 = c(100,
    100, 100), maxi = T)

Maximization Problem with Objective Function Coefficients
x1 x2 x3
22 30 25


Optimal solution has the following values
      x1       x2       x3
33.33333 16.66667 16.66667
The optimal value of the objective  function is 1650.
> 
```

Interpretation:

We learnt the R coding for solving LPP by simplex method.

The optimal value of the objective function is **Max Z= 1650** with the solution.

$$\mathbf{x_1 = 33.34,\; x_2 = 16.67,\; x_3 = 16.67}$$

Big M - Method

Exercise: 38

Date: 20.02.2020

Solve the following LPP by Big M method, Max Z = $5xx_1 - 6x_2 - 7x_3$

Subject to the constraints $x_1 + 5x_2 - 3x_3 \geq 15$

$$5x_1 - 6x_2 + 10x_3 \geq 0$$

$$x_1 + x_2 + x_3 = 5,\ x_1, x_2, x_3 \geq 0$$

Aim: To create R coding to solve the LPP by Big M method.

R code & Output:

```
>
> #-------------------BIG M METHOD-------------------#
> library(boot)
> objective<-c(5,-6,-7)
> cons1<-c(1,5,-3)
> cons2<-c(5,-6,10)
> cons3<-c(1,1,1)
> simplex(a=objective,A1=rbind(cons1,cons2),b1=c(15,0),A3=cons3,b3=5,maxi=T)

Linear Programming Results

Call : simplex(a = objective, A1 = rbind(cons1, cons2), b1 = c(15, 0),
    A3 = cons3, b3 = 5, maxi = T)

Maximization Problem with Objective Function Coefficients
x1 x2 x3
 5 -6 -7


Optimal solution has the following values
      x1       x2       x3
2.727273 2.272727 0.000000
The optimal value of the objective  function is 0.
>
```

Interpretation:

We learnt the R coding for solving LPP by Big M method.

The optimal value of the objective function is **Max Z= 0** with the solution.

$$\mathbf{x_1 = 2.7,\ x_2 = 2.27,\ x_3 = 0}$$

TWO PHASE SIMPLEX METHOD

Exercise: 39

Date: 21.02.2020

Solve the following LPP by two phase simplex method:

$$\text{Max } Z = 12x_1 + 15x_2 + 9x_3$$

Subject to the constraints

$8x_1 + 16x_2 + 12x_3 \leq 250$

$4x_1 + 8x_2 + 10x_3 \geq 80$

$7x_1+9x_2+8x_3 = 10$, $x_1, x_2, x_3 \geq 0$

Aim:

To create r coding to solve the LPP by Two phase simplex method.

R Code and Output:

```
> #--------TWO PHASE METHOD-----------#
> library(boot)
> objective<-c(12,15,9)
> cons1<-c(8,16,12)
> cons2<-c(4,8,10)
> cons3<-c(7,9,8)
> simplex(a=objective,A1=cons1,b1=250,A2=cons2,b2=80,A3=cons3,b3=105,maxi=T)

Linear Programming Results

Call : simplex(a = objective, A1 = cons1, b1 = 250, A2 = cons2, b2 = 80,
    A3 = cons3, b3 = 105, maxi = T)

Maximization Problem with Objective Function Coefficients
x1 x2 x3
12 15  9


Optimal solution has the following values
x1 x2 x3
 6  7  0
The optimal value of the objective  function is 177.
```

Interpretation:

We learned R coding for **solving LPP by Two phase simplex method.**

The optimal value of the objective function is Max Z = 177 with the solution

$$x_1 = 6,\ x_2 = 7,\ x_3 = 0$$

Question:

A company produces two models of chair: 4P and 3P. The Model 4P needs 4 legs, 1 seat and back. On the other hand, the model 3P needs 3 legs and 1 seat. The company has an initial stock of 200 legs, 500 seats and 100 backs. If the company needs more legs, seats and backs, it can buy standard wood blocks, whose cost is 80 Rs per block. The company can produce 10 seats, 20 legs and 2 backs from a standard wood block. The cost of producing the model 4P is 30 Rs/chair, meanwhile the cost of producing the model 3P is 40 Rs/chair. Finally, the company informs that the minimum number of chairs to produce is 1000 units per month. Define a linear programming model, which minimizes the total cost (the production costs of the two chairs, plus the buying of new wood blocks).

Transportation Problem

Exercise: 40

Date: 24.02.2020

Solve the following transportation problem,

Origins	Destinations				Supply
	1	2	3	4	
A	15	12	42	33	23
B	80	42	26	81	44
C	90	40	66	60	33
Demand	23	31	16	30	100

Aim:

To create R coding to solve the Transportation problem.

R code & Output:

```
> #-------------------Transportation problem-------------------#
> cost<-matrix(c(15,12,42,33,80,42,26,81,90,40,66,60),nrow=3,ncol=4,byrow=T)
> rowsigns<-rep("=",3)
> rowvalues<-c(23,44,33)
> colsigns<-rep("=",4)
> colvalues<-c(23,31,16,30)
> fit<-lp.transport(cost,"min",rowsigns,rowvalues,colsigns,colvalues)
> fit$solution
     [,1] [,2] [,3] [,4]
[1,]   23    0    0    0
[2,]    0   28   16    0
[3,]    0    3    0   30
> fit
Success: the objective function is 3857
> 
```

Interpretation:

We learnt the R coding for solving Transportation problem.

The optimal allocations for the given transportation problem is as follows,

Origin	Destination	Number of units allotted
A	1	23
B	2	28
B	3	16
C	2	3
C	4	30

The minimum transportation cost is **Rs.3857.**

Assignment Problem

Exercise: 41

Date: 25.02.2020

Five jobs have to be allotted to five machines, such that the total time taken to perform all the jobs are minimum. Make the optimum allotment of the jobs to the machines.

Machine \ Job	I	II	III	IV	V
A	11	17	18	16	20
B	9	7	12	6	15
C	13	16	15	12	16
D	21	24	17	28	26
E	14	10	12	11	15

Aim:

To create R coding to solve the assignment problem.

R code & Output:

```
> #-------------------Assignment problem-------------------#
> time<-matrix(c(11,17,18,16,20,09,7,12,6,15,13,16,15,12,16,21,24,17,28,26,14,10,12,11,15),nrow=5,ncol=5,byrow=T)
> fit<-lp.assign(time,direction="min")
> fit$solution
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    0    0    1    0
[3,]    0    0    0    0    1
[4,]    0    0    1    0    0
[5,]    0    1    0    0    0
> fit
Success: the objective function is 60
```

Interpretation:

We learnt the R coding for solving Assignment problem.

The optimal allocations for the given assignment problem is as follows,

Job	Machine	Time to complete the job
I	A	11
II	E	10
III	D	17
IV	B	6
V	C	16

The minimum time to complete the job is **60.**

$\overline{X}$ and R chart

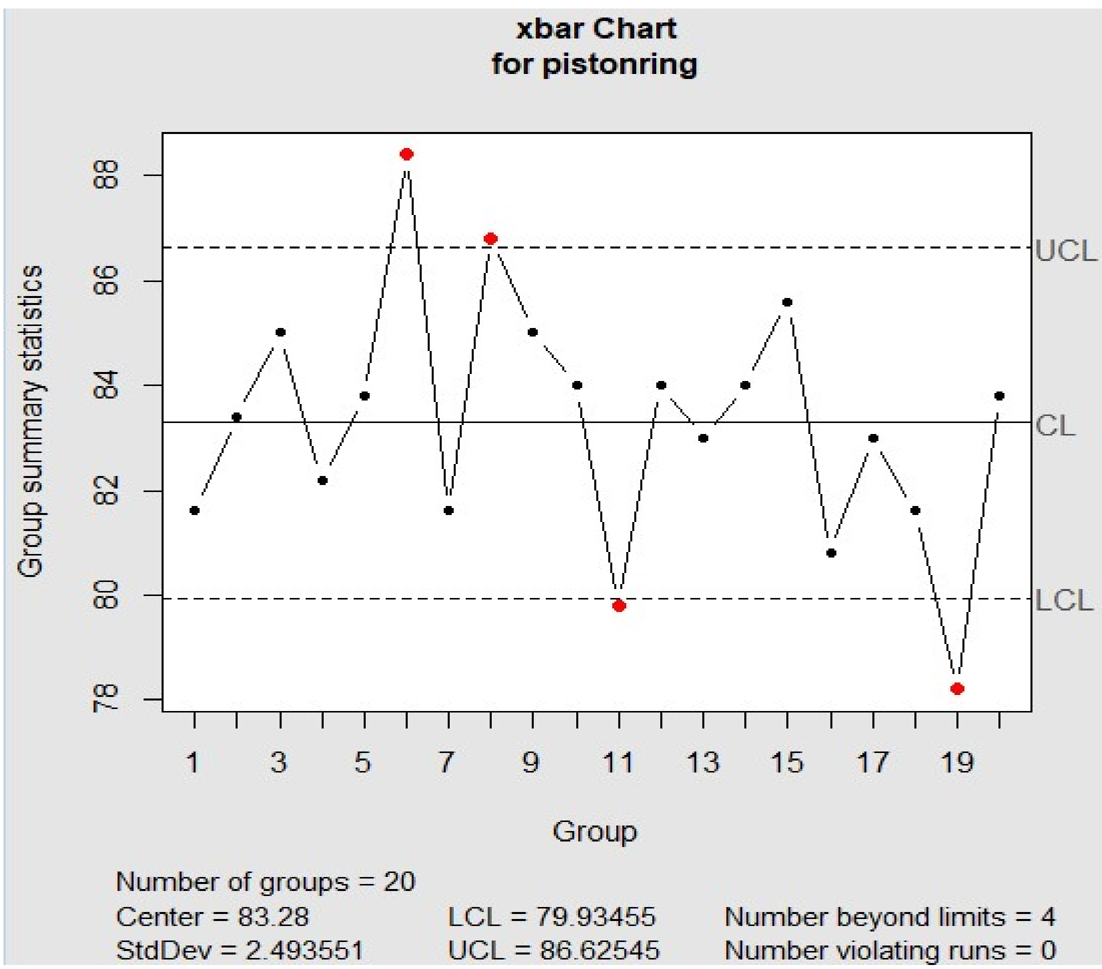
Exercise: 42

Date: 27.02.2020

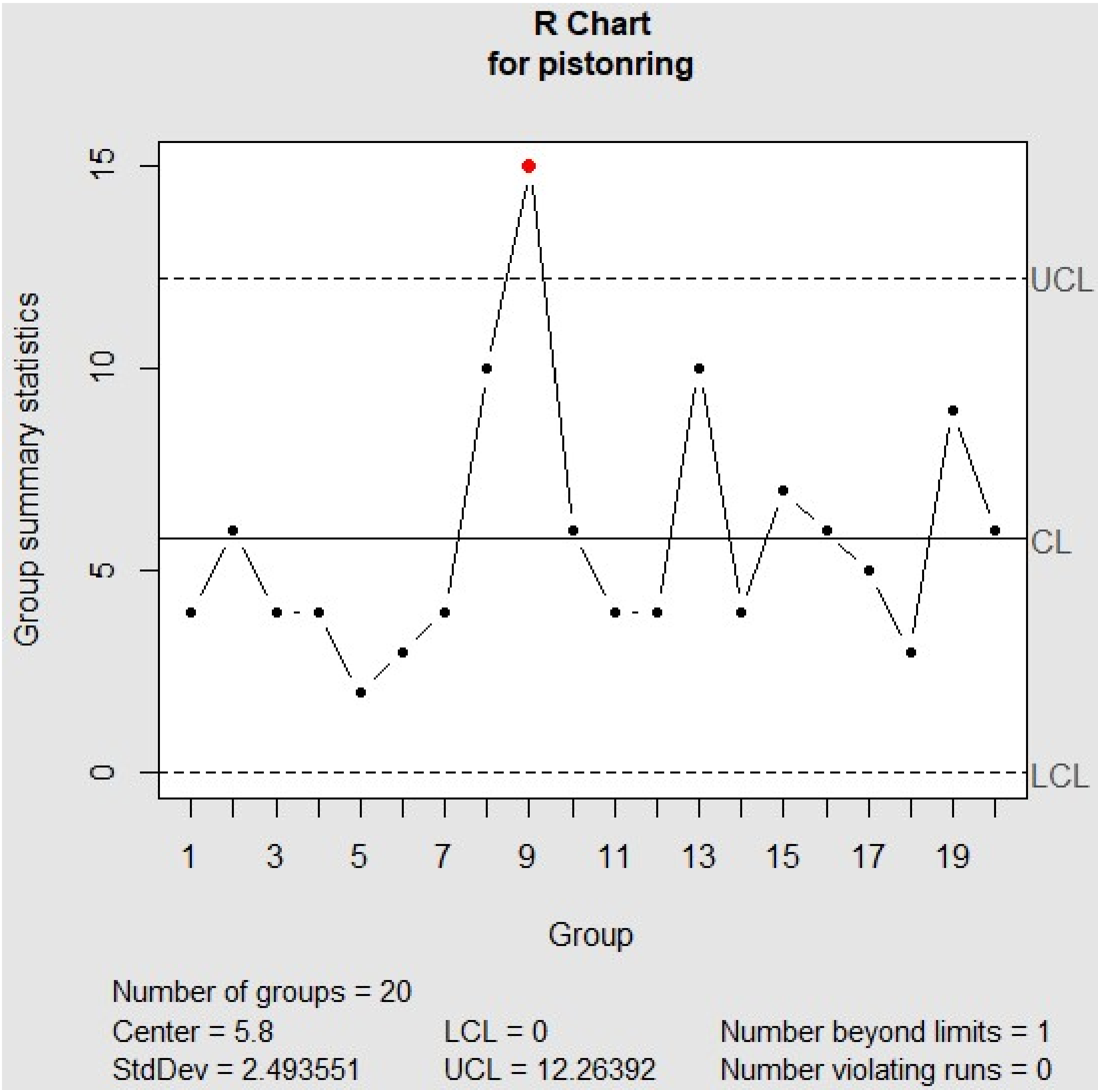
Construct $\overline{X}$ and R chart for the following data.

Sample	X1	X2	X3	X4	X5
1	83	79	81	82	83
2	83	81	85	87	81
3	85	87	83	84	86
4	80	81	83	84	83
5	83	84	85	83	84
6	88	87	89	90	88
7	80	81	82	84	81
8	79	89	88	89	89
9	78	83	85	86	93
10	88	83	82	85	82
11	78	80	78	82	81
12	81	85	85	85	84
13	77	82	84	85	87
14	81	85	85	85	84
15	85	87	82	85	89
16	83	83	77	81	80
17	85	84	84	80	82
18	82	83	80	80	83
19	75	77	84	77	78
20	85	85	86	83	80

```
> qc<-read.table("U:\\ugst46\\x-bar and R charrt data.csv",header=T,sep=",")
> qc
   Sample X1 X2 X3 X4 X5
1       1 83 79 81 82 83
2       2 83 81 85 87 81
3       3 85 87 83 84 86
4       4 80 81 83 84 83
5       5 83 84 85 83 84
6       6 88 87 89 90 88
7       7 80 81 82 84 81
8       8 79 89 88 89 89
9       9 78 83 85 86 93
10     10 88 83 82 85 82
11     11 78 80 78 82 81
12     12 81 85 85 85 84
13     13 77 82 84 85 87
14     14 81 85 85 85 84
15     15 85 87 82 85 89
16     16 83 83 77 81 80
17     17 85 84 84 80 82
18     18 82 83 80 80 83
19     19 75 77 84 77 78
20     20 85 85 86 83 80
```



```
> pistonring<-qc[,-1]
> #------------------------X Bar Chart----------------------------#
> qcc(pistonring,type="xbar",nsigma=3)
List of 11
 $ call      : language qcc(data = pistonring, type = "xbar", nsigmas = 3)
 $ type      : chr "xbar"
 $ data.name : chr "pistonring"
 $ data      : int [1:20, 1:5] 83 83 85 80 83 88 80 79 78 88 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 81.6 83.4 85 82.2 83.8 88.4 81.6 86.8 85 84 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : int [1:20] 5 5 5 5 5 5 5 5 5 5 ...
 $ center    : num 83.3
 $ std.dev   : num 2.49
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 79.9 86.6
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
> #------------------------R Chart-------------------------------#
> qcc(pistonring,type="R",nsigma=3)
List of 11
 $ call      : language qcc(data = pistonring, type = "R", nsigmas = 3)
 $ type      : chr "R"
 $ data.name : chr "pistonring"
 $ data      : int [1:20, 1:5] 83 83 85 80 83 88 80 79 78 88 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named int [1:20] 4 6 4 4 2 3 4 10 15 6 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : int [1:20] 5 5 5 5 5 5 5 5 5 5 ...
 $ center    : num 5.8
 $ std.dev   : num 2.49
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 0 12.3
```



Interpretation:

We learnt the R coding to construct X bar and R chart for the given data.

1. The control limit for X-bar chart are LCL=79.93455 and UCL=86.665. From the control chart, the four sampling observation 6,8,11 and 19 are falling outside of the UCL and LCL. So we conclude that the process is out of control.
2. The control limit for R chart are LCL=0 and UCL=12.26392. From the control chart, the 9th sampling observation falling outside of the LCL and UCL. So we conclude that the process is out of control.

$\overline{X}$ and S chart

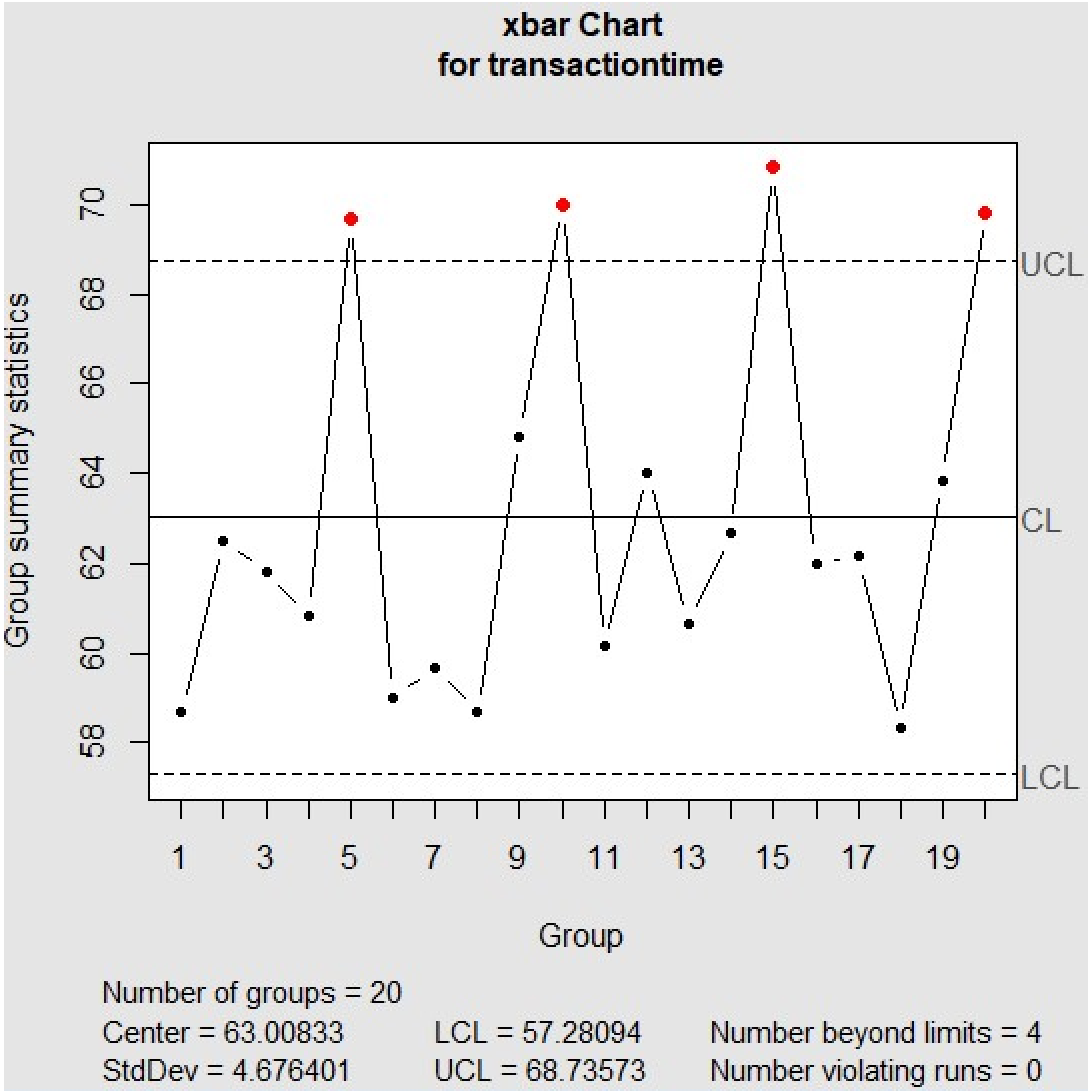
Exercise: 43

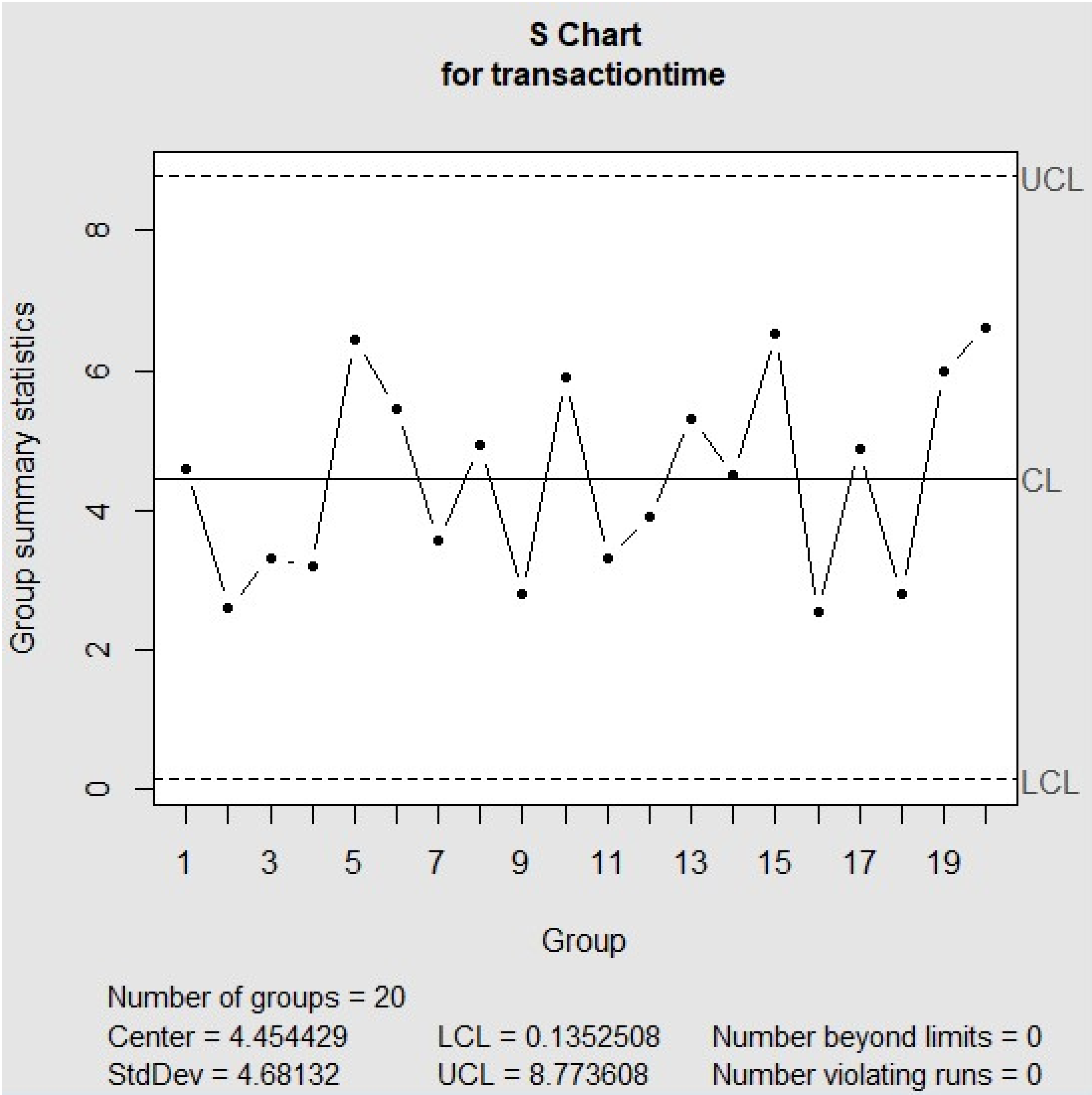
Date: 27.02.2020

Construct $\overline{X}$ and S chart for the following data.

Sample	X1	X2	X3	X4	X5	X6
1	63	55	56	53	61	64
2	60	63	60	65	61	66
3	57	60	61	65	66	62
4	58	64	60	61	57	65
5	79	68	65	61	74	71
6	55	66	62	63	56	52
7	57	61	58	64	55	63
8	58	51	61	57	66	59
9	65	66	62	68	61	67
10	73	66	61	70	72	78
11	57	63	56	64	62	59
12	66	63	65	59	70	61
13	63	53	69	60	61	58
14	68	67	59	58	65	59
15	70	62	66	80	71	76
16	65	59	60	61	62	65
17	63	69	58	56	66	61
18	61	56	62	59	57	55
19	65	57	69	62	58	72
20	70	60	67	79	75	68

```
>
> #-------------------------X Bar Chart----------------------------#
> qcc(transactiontime,type="xbar",nsigma=3)
List of 11
 $ call      : language qcc(data = transactiontime, type = "xbar", nsigmas = 3)
 $ type      : chr "xbar"
 $ data.name : chr "transactiontime"
 $ data      : int [1:20, 1:6] 63 60 57 58 79 55 57 58 65 73 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 58.7 62.5 61.8 60.8 69.7 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : int [1:20] 6 6 6 6 6 6 6 6 6 6 ...
 $ center    : num 63
 $ std.dev   : num 4.68
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 57.3 68.7
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
> #-------------------------S Chart--------------------------------#
> qcc(transactiontime,type="S",nsigma=3)
List of 11
 $ call      : language qcc(data = transactiontime, type = "S", nsigmas = 3)
 $ type      : chr "S"
 $ data.name : chr "transactiontime"
 $ data      : int [1:20, 1:6] 63 60 57 58 79 55 57 58 65 73 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 4.59 2.59 3.31 3.19 6.44 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : int [1:20] 6 6 6 6 6 6 6 6 6 6 ...
 $ center    : num 4.45
 $ std.dev   : num 4.68
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 0.135 8.774
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
>
```


xbar Chart
for transactiontime
Group summary statistics
58
60
62
64
66
68
70
UCL
CL
LCL
1
3
5
7
9
11
13
15
17
19
Group
Number of groups = 20
Center = 63.00833
StdDev = 4.676401
LCL = 57.28094
UCL = 68.73573
Number beyond limits = 4
Number violating runs = 0



Interpretation:

We learnt the R coding to construct X bar and S chart for the given data.

1. The control limit for X-bar chart are **LCL=57.28** and **UCL=68.73**. From the control chart, the four sampling observation 5,10,15 and 15th are falling outside of the UCL and LCL. So we conclude that the process is out of control.
2. The control limit for R chart are **LCL=0.13** and **UCL=8.77**. From the control chart all the sampling observation falling inside of the LCL and UCL. So we conclude that the process is in control.

np Chart

Exercise: 44

Date: 28.02.2020

Consider the data on Number of defectives in sample of 1000 ceramic substances.

Sample	1	2	3	4	5	6	7	8	9	10
Defectives	33	37	21	39	18	20	35	41	33	37
n	100	100	100	100	100	100	100	100	100	100

Sample	11	12	13	14	15	16	17	18	19	20
Defectives	25	41	24	30	31	19	35	27	15	19
n	100	100	100	100	100	100	100	100	100	100

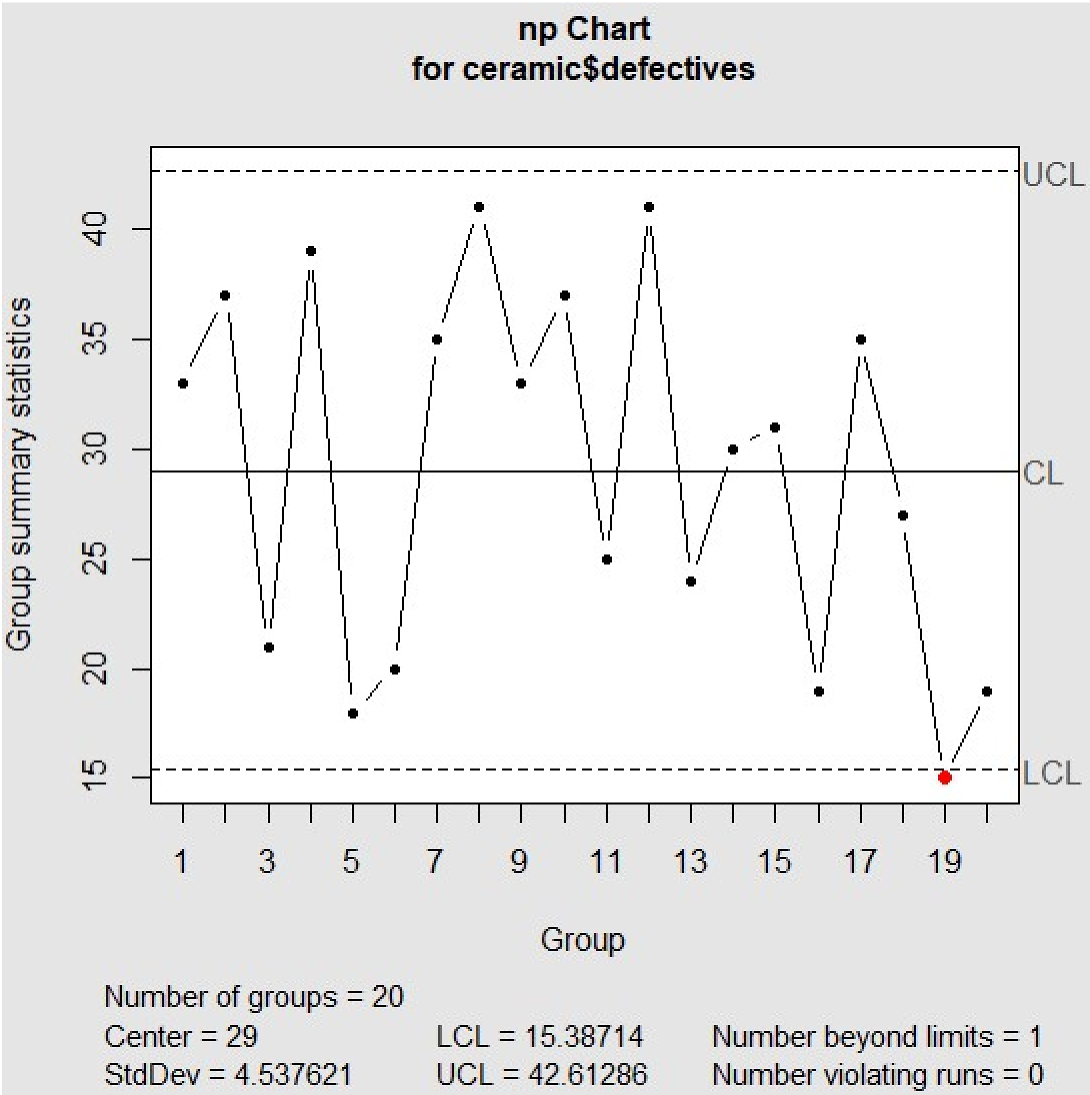
Construct control chart for number of defectives based on the given data.

Aim:

To create R coding to construct np chart for the given data.

R code and Output:

```
> #---------------------------np chart--------------------------------#
> defectives<-c(33,37,21,39,18,20,35,41,33,37,25,41,24,30,31,19,35,27,15,19)
> samplesize<-rep(100,20)
> ceramic<-data.frame(defectives,samplesize)
> qcc(ceramic$defectives, sizes=ceramic$samplesize,type="np")
List of 11
 $ call      : language qcc(data = ceramic$defectives, type = "np", sizes = ceramic$samplesize)
 $ type      : chr "np"
 $ data.name : chr "ceramic$defectives"
 $ data      : num [1:20, 1] 33 37 21 39 18 20 35 41 33 37 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 33 37 21 39 18 20 35 41 33 37 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : num [1:20] 100 100 100 100 100 100 100 100 100 100 ...
 $ center    : num 29
 $ std.dev   : num 4.54
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 15.4 42.6
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
> 
```



Interpretation:

We learnt the R coding for construction of np chart for the given data.

The control limit for np chart are **LCL = 15.387** and **UCL = 42.612**. From the control chart, the 19^{th} sampling observation is falling outside of the UCL and LCL. So we conclude that the process is out of control.

P Chart

Exercise: 45

Date: 2.03.2020

Consider the data on Number of defectives in the 20 independent samples of varying sizes from production process.

Sample	1	2	3	4	5	6	7	8	9	10
Defectives	33	37	21	39	18	20	35	41	33	37
n	80	95	65	90	85	80	75	80	90	90

Sample	11	12	13	14	15	16	17	18	19	20
Defectives	25	41	24	30	31	19	35	27	15	19
n	100	110	85	95	85	95	100	110	85	90

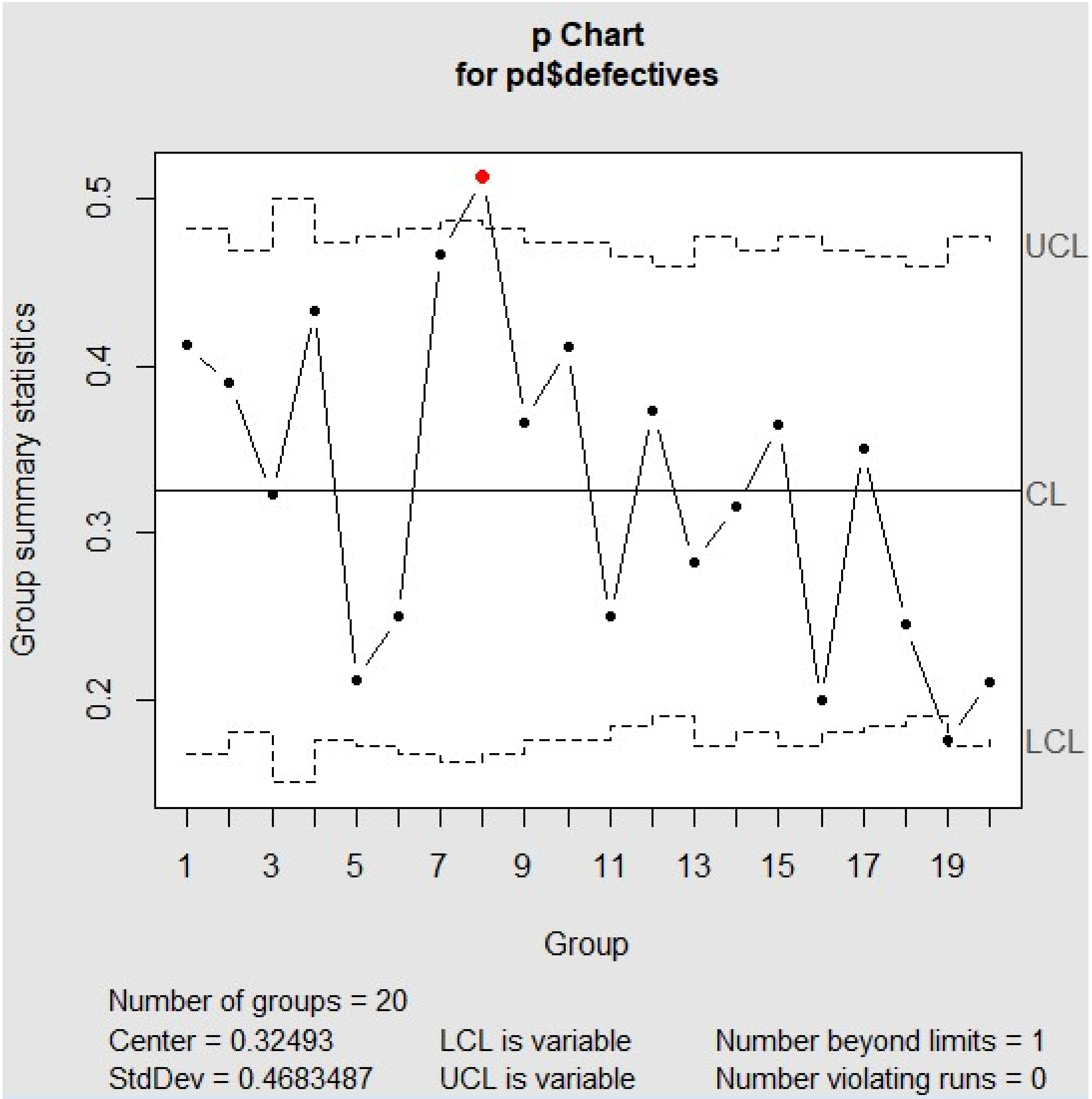
Construct control chart for number of defectives based on the given data.

Aim:

To create R coding to construct p chart for the given data.

R code and Output:

```
> library(qcc)
> #--------------------------p chart--------------------------------#
> defectives<-c(33,37,21,39,18,20,35,41,33,37,25,41,24,30,31,19,35,27,15,19)
> samplesize<-c(80,95,65,90,85,80,75,80,90,90,100,110,85,95,85,95,100,110,85,90)
> pd<-data.frame(defectives,samplesize)
> qcc(pd$defectives, sizes=pd$samplesize,type="p")
List of 11
 $ call      : language qcc(data = pd$defectives, type = "p", sizes = pd$samplesize)
 $ type      : chr "p"
 $ data.name : chr "pd$defectives"
 $ data      : num [1:20, 1] 33 37 21 39 18 20 35 41 33 37 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 0.412 0.389 0.323 0.433 0.212 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : num [1:20] 80 95 65 90 85 80 75 80 90 90 ...
 $ center    : num 0.325
 $ std.dev   : num 0.468
 $ nsigmas   : num 3
 $ limits    : num [1:20, 1:2] 0.168 0.181 0.151 0.177 0.173 ...
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
```



Interpretation:

We learnt the R coding for construction of **p chart** for the given data.

From the control chart, the 8^{th} sampling observation is falling out side of the UCL. So we conclude that the process is out of control.

c Chart

Exercise: 46

Date: 2.03.2020

Number of defects in samples of five Printed circuit Boards is given below. Construct control chart for total number of defects in a sample and interpret.

Sample	1	2	3	4	5	6	7	8	9	10
Defectives	6	4	8	10	9	12	16	2	3	10
n	50	50	50	50	50	50	50	50	50	50

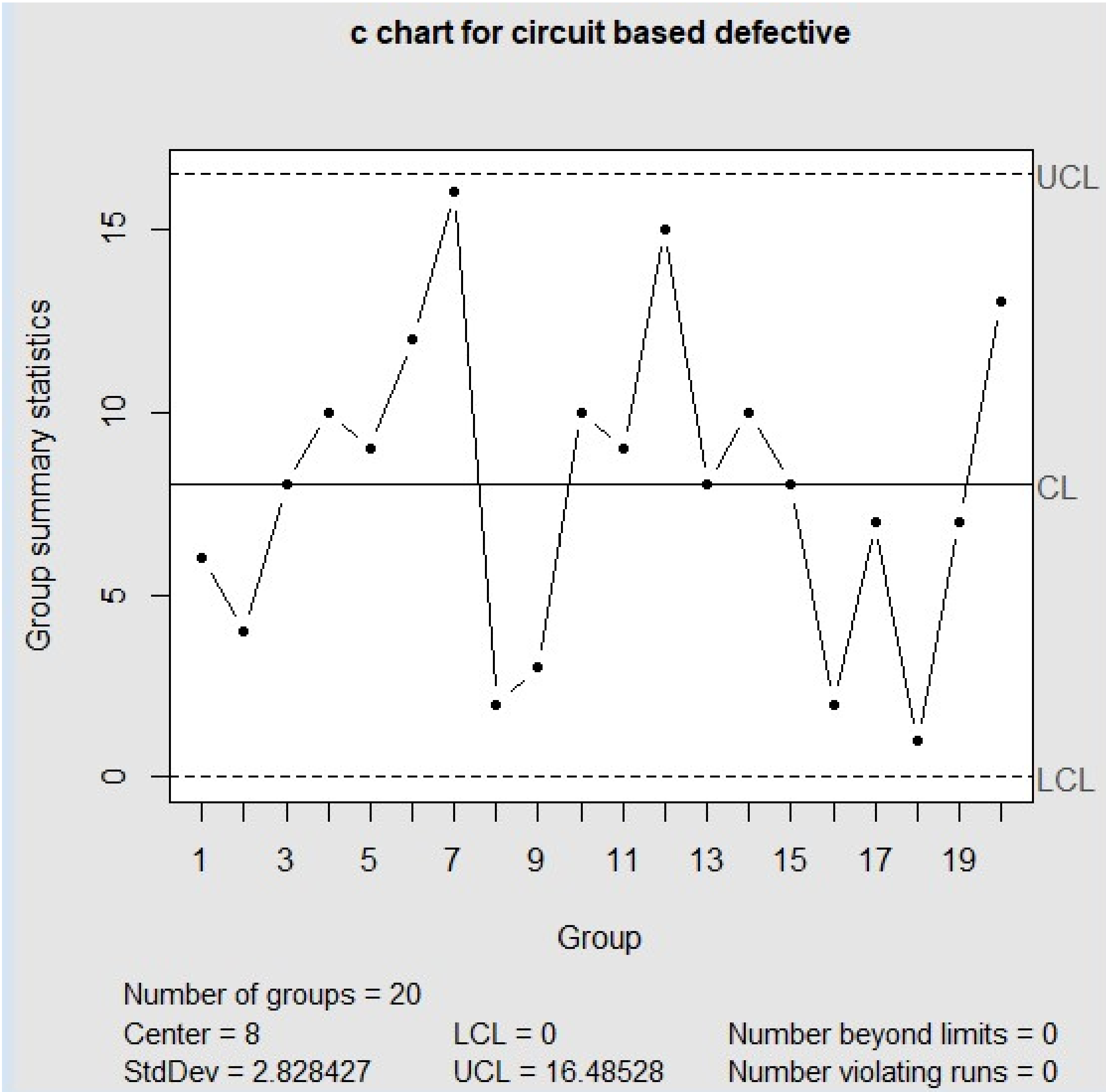
Sample	11	12	13	14	15	16	17	18	19	20
Defectives	9	15	8	10	8	2	7	1	7	13
n	50	50	50	50	50	50	50	50	50	50

AIM:

To create R coding to construct c chart for the given data.

R code and Output:

```
> library(qcc)
> #--------------------------c chart--------------------------------#
> defects<-c(6,4,8,10,9,12,16,2,3,10,9,15,8,10,8,2,7,1,7,13)
> samplesize_1<-c(rep(50,20))
> circuit<-data.frame(defects,samplesize_1)
> qcc(circuit$defects, sizes=circuit$samplesize,type="c", title = "c chart for circuit based defective")
List of 11
 $ call      : language qcc(data = circuit$defects, type = "c", sizes = circuit$samplesize, title = "c chart for circuit based defective")
 $ type      : chr "c"
 $ data.name : chr "circuit$defects"
 $ data      : num [1:20, 1] 6 4 8 10 9 12 16 2 3 10 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 6 4 8 10 9 12 16 2 3 10 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : num [1:20] 50 50 50 50 50 50 50 50 50 50 ...
 $ center    : num 8
 $ std.dev   : num 2.83
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 0 16.5
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
```



Interpretation:

We learnt the R coding to construct c chart for the given data.

The control limits for c chart are **LCL = 0** and **UCL =16.485**. From the control chart, all the points are falling within the control limits and we've found no pattern or violating run. So we conclude that the process is in control.

U Chart

Exercise: 47

Date: 2.03.2020

Number of defects in samples of five Printed circuit Boards is given below. Construct control chart for total number of defects per unit in a sample and interpret.

Sample	1	2	3	4	5	6	7	8	9	10
Defectives	6	4	8	10	9	12	16	2	3	10
n	50	50	50	50	50	50	50	50	50	50

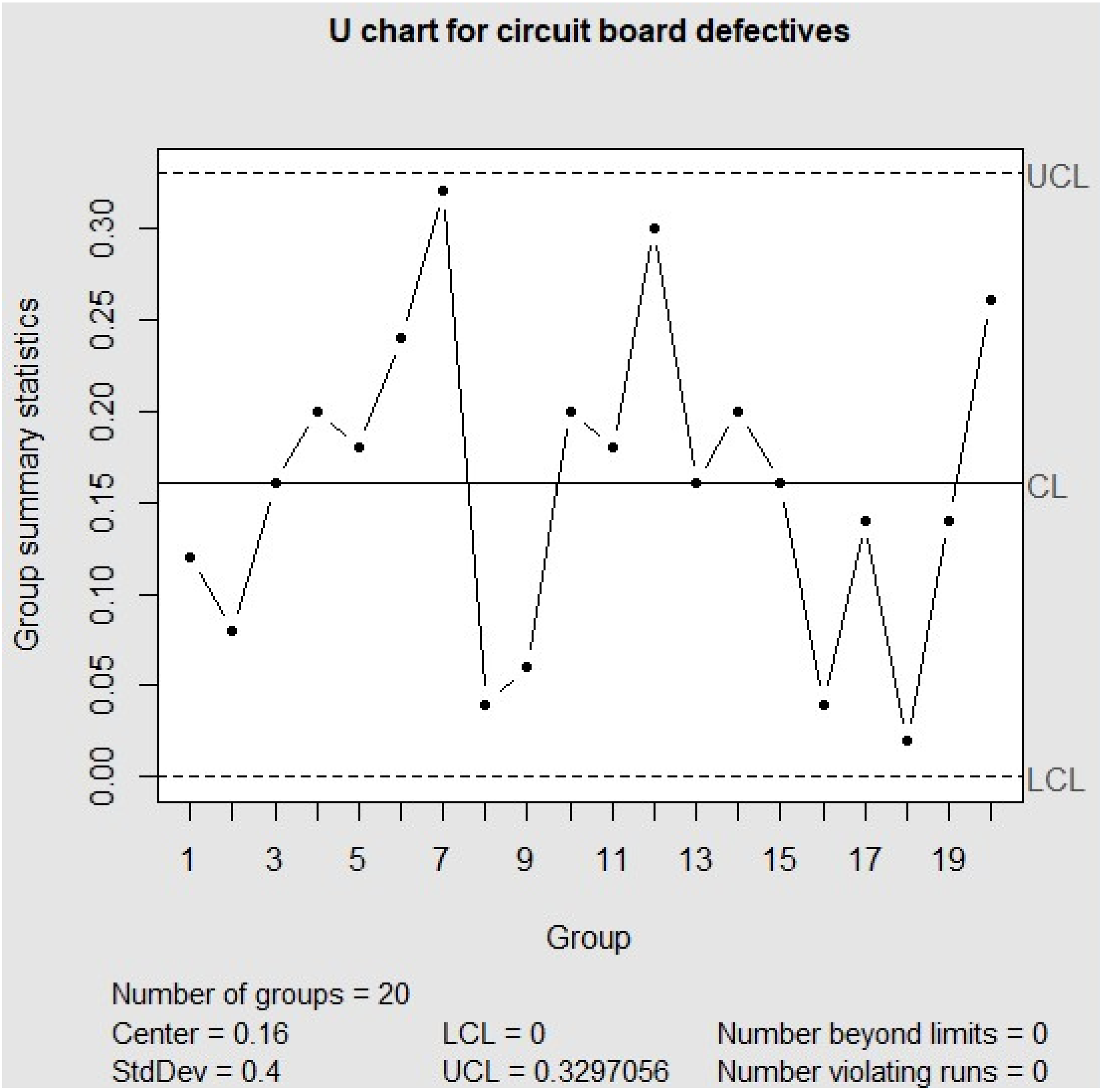
Sample	11	12	13	14	15	16	17	18	19	20
Defectives	9	15	8	10	8	2	7	1	7	13
n	50	50	50	50	50	50	50	50	50	50

Aim:

To create R coding to construct u chart for the given data.

R code and Output:

```
> library(qcc)
> #-------------------------U chart-------------------------------#
> defects<-c(6,4,8,10,9,12,16,2,3,10,9,15,8,10,8,2,7,1,7,13)
> samplesize_1<-c(rep(50,20))
> circuit<-data.frame(defects,samplesize_1)
> qcc(circuit$defects, sizes=circuit$samplesize,type="u", title = "U chart for circuit board defectives")
List of 11
 $ call      : language qcc(data = circuit$defects, type = "u", sizes = circuit$samplesize, title = "U chart for circuit board defectives")
 $ type      : chr "u"
 $ data.name : chr "circuit$defects"
 $ data      : num [1:20, 1] 6 4 8 10 9 12 16 2 3 10 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:20] 0.12 0.08 0.16 0.2 0.18 0.24 0.32 0.04 0.06 0.2 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ...
 $ sizes     : num [1:20] 50 50 50 50 50 50 50 50 50 50 ...
 $ center    : num 0.16
 $ std.dev   : num 0.4
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 0 0.33
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
```



Interpretation:

We learnt the R coding to construct U chart for the given data.

The control limits for u chart are **LCL = 0** and **UCL =0.3297**. From the control chart, all the points are falling within the control limits and we've found no pattern or violating run. So we conclude that the process is in control.

COMPARISON OF SHEWART AND CUSUM CHART

Exercise: 48

Date: 4.03.2020

For the following quality characteristics construct Shewart and Cusum control chart and give your interpretations

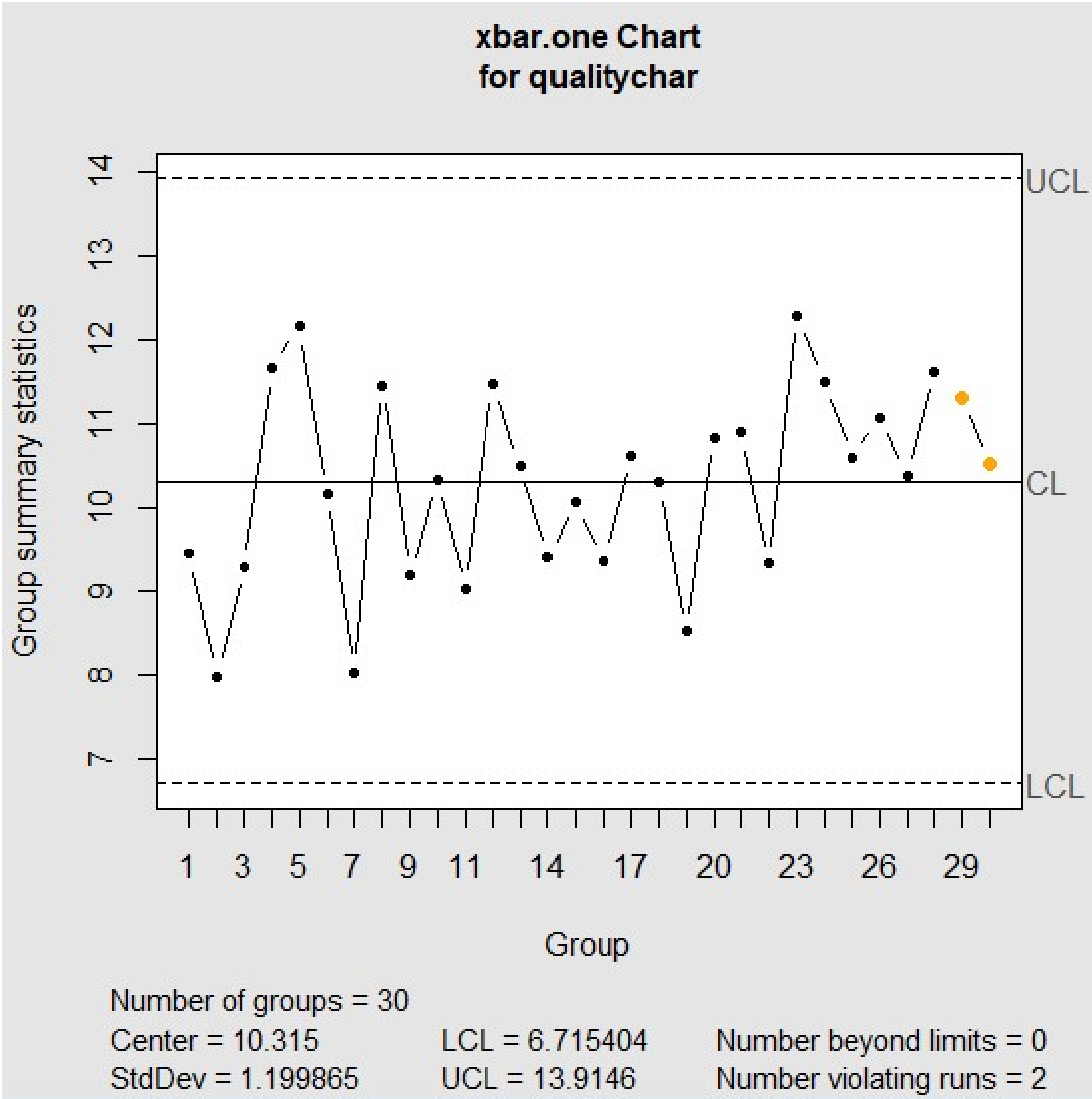
9.45	7.99	9.29	11.66	12.16	10.18	8.04	11.46	9.2	10.34
9.03	11.47	10.51	9.4	10.08	9.37	10.62	10.31	8.52	10.84
10.9	9.33	12.29	11.5	10.6	11.08	10.38	11.62	11.31	10.52

Aim:

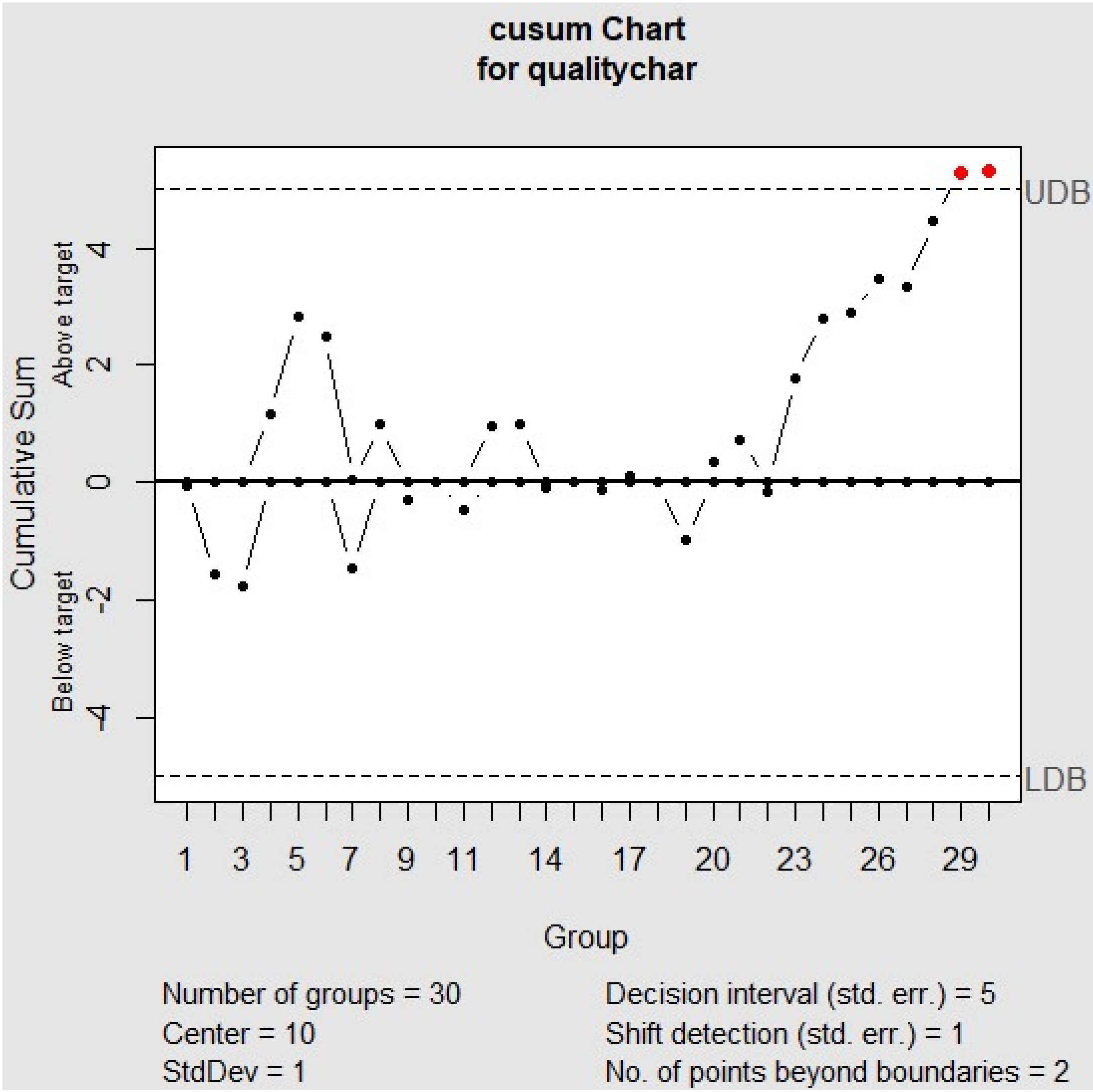
To create R coding to compare the shewart and cusum control chart for the given data.

R code and Output:

```
> #Shewart and cusum#
> library(qcc)
> qualitychar<-c(9.45,7.99,9.29,11.66,12.16,10.18,8.04,11.46,9.2,10.34,9.03,11.47,10.51,9.4,10.08,9.37,10.62,10.31,8.52,10.84,10.9
+ ,9.33,12.29,11.5,10.6,11.08,10.38,11.62,11.31,10.52)
> qcc(qualitychar, type="xbar.one",nsigma=3)
List of 11
 $ call      : language qcc(data = qualitychar, type = "xbar.one", nsigmas = 3)
 $ type      : chr "xbar.one"
 $ data.name : chr "qualitychar"
 $ data      : num [1:30, 1] 9.45 7.99 9.29 11.66 12.16 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics: Named num [1:30] 9.45 7.99 9.29 11.66 12.16 ...
  ..- attr(*, "names")= chr [1:30] "1" "2" "3" "4" ...
 $ sizes     : int [1:30] 1 1 1 1 1 1 1 1 1 1 ...
 $ center    : num 10.3
 $ std.dev   : num 1.2
 $ nsigmas   : num 3
 $ limits    : num [1, 1:2] 6.72 13.91
  ..- attr(*, "dimnames")=List of 2
 $ violations:List of 2
 - attr(*, "class")= chr "qcc"
```




```
> cusum(qualitychar, decision.interval= 5, std.dev=1, center = 10, sizes = 1)
List of 14
 $ call             : language cusum(data = qualitychar, sizes = 1, center = 10, std.dev = 1, decision.interval = 5
 $ type             : chr "cusum"
 $ data.name        : chr "qualitychar"
 $ data             : num [1:30, 1] 9.45 7.99 9.29 11.66 12.16 ...
  ..- attr(*, "dimnames")=List of 2
 $ statistics       : Named num [1:30] 9.45 7.99 9.29 11.66 12.16 ...
  ..- attr(*, "names")= chr [1:30] "1" "2" "3" "4" ...
 $ sizes            : num [1:30] 1 1 1 1 1 1 1 1 1 1 ...
 $ center           : num 10
 $ std.dev          : num 1
 $ pos              : num [1:30] 0 0 0 1.16 2.82 ...
 $ neg              : num [1:30] -0.05 -1.56 -1.77 0 0 ...
 $ head.start       : num 0
 $ decision.interval: num 5
 $ se.shift         : num 1
 $ violations       :List of 2
 - attr(*, "class")= chr "cusum.qcc"
>
```



Interpretation:

We learnt the R coding to compare **the Shewart and Cusum control chart** for the given data.

The shewart control chart identifies a possible shift in the process based on the runs but still it does not provide strong evidence. The Cusum control chart provides evidence of process shift based on points lying outside upper control limits.

EXPONENTIAL MOVING AVERAGE CONTROL CHART

Exercise: 49

Date: 6.03.2020

For the following quality characteristics construct Exponential Moving average control chart and give your interpretation.

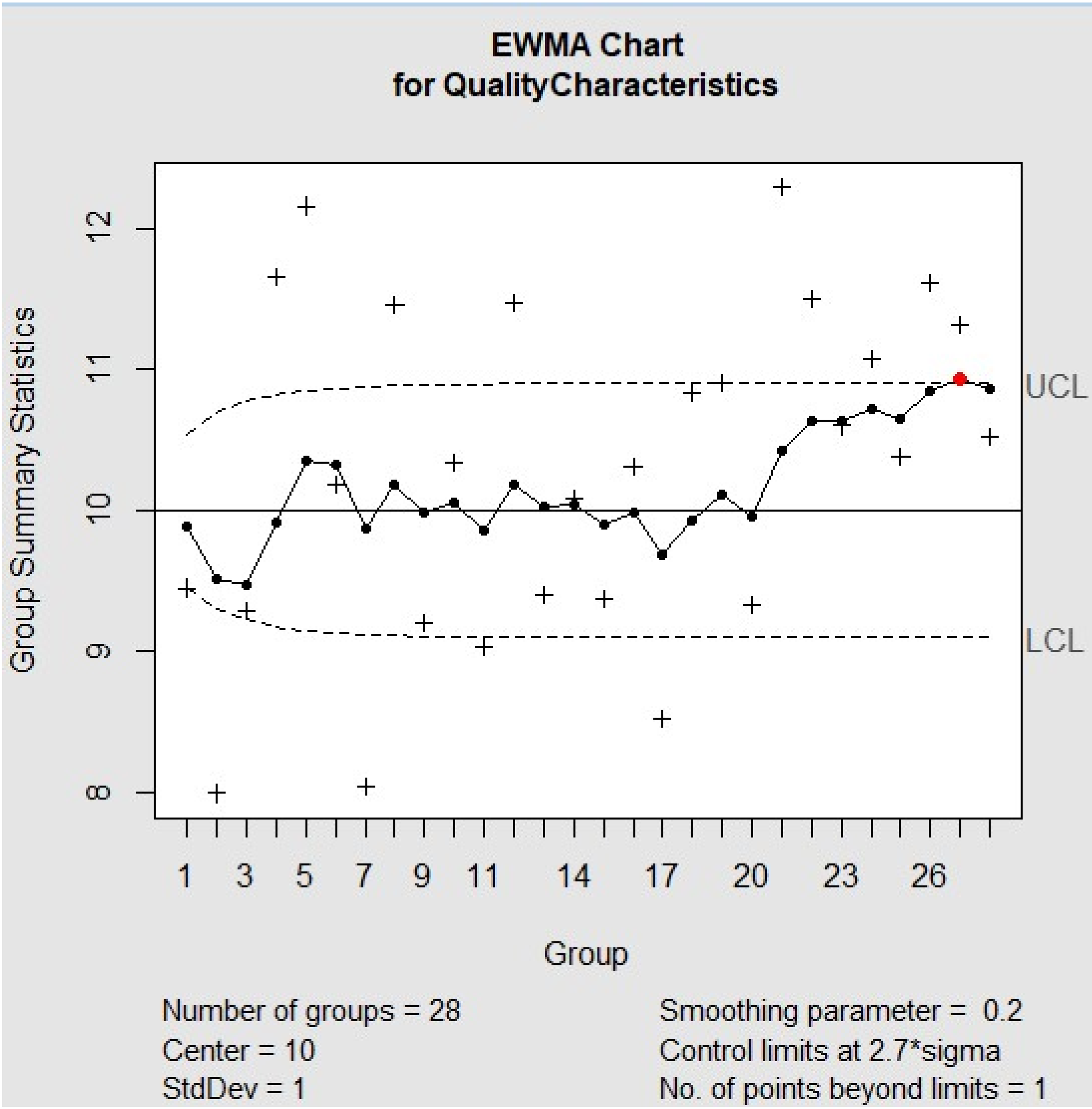
9.45	7.99	9.29	11.66	12.16	10.18	8.04	11.46	9.2	10.34
9.03	11.47	10.51	9.4	10.08	9.37	10.62	10.31	8.52	10.9
9.33	12.29	11.5	10.6	11.08	10.38	11.62	11.31	10.52	

Aim:

To create R codings to construct exponential moving average control charts for the given data

R Code & Output

```
> #-----Exponential Moving Average Chart-----#
> library(qcc)
  __ _  ___ ___
 / _` |/ __/ __|  Quality Control Charts and
| (_| | (_| (__   Statistical Process Control
 \__  |\___\___|
    |_|           version 2.7
Type 'citation("qcc")' for citing this R package in publications.
> QualityCharacteristics<-c(9.45,7.99,9.29,11.66,12.16,10.18,8.04,11.46,9.2,10.34,
+ 9.03,11.47,9.4,10.08,9.37,10.31,8.52,10.84,10.9,9.33,12.29,11.5,10.6,11.08,
+ 10.38,11.62,11.31,10.52)
> Quality<-ewma(QualityCharacteristics,lamda=0.1,nsigma=2.7,std.dev=1,center=10)
> 
```



Interpretation:

We learned R codings for exponential weighted moving average control charts to the given data.

EWMA control chart provides evidence for shift in process mean for the given characteristics and hence further inspection of the person.

PROCESS CAPABILITYANALYSIS -I

Exercise: 50

Date: 6.03.2020

Construct Cp,Cpl, Cpu, Cpk and Cpm for the following data and interpret on process control.

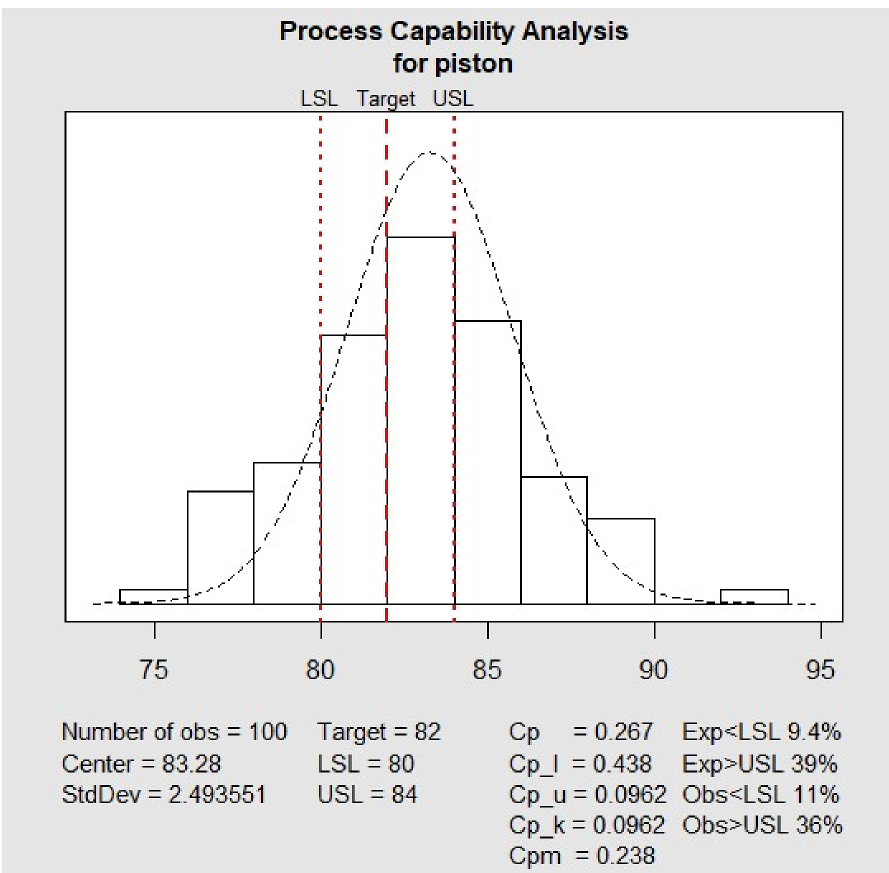
Sample	X1	X2	X3	X4	X5
1	83	79	81	82	83
2	83	81	85	87	81
3	85	87	83	84	86
4	80	81	83	84	83
5	83	84	85	83	84
6	88	87	89	90	88
7	80	81	82	84	81
8	79	89	88	89	89
9	78	83	85	86	93
10	88	83	82	85	82
11	78	80	78	82	81
12	81	85	85	85	84
13	77	82	84	85	87
14	81	85	85	85	84
15	85	87	82	85	89
16	83	83	77	81	80
17	85	84	84	80	82
18	82	83	80	80	83
19	75	77	84	77	78
20	85	85	86	83	80

Aim:

To create r coding to construct process capability analysis control charts for the given data

R Code and Output:

```
> library(qcc)
> data<-file.choose()
> datachart<-read.csv(data)
> piston<-datachart[,-1]
> q<-qcc(piston,type="xbar",nsigmas=3,plot=F)
> process.capability(q,spec.limits=c(80,84))

Process Capability Analysis

Call:
process.capability(object = q, spec.limits = c(80, 84))

Number of obs = 100          Target = 82
       Center = 83.28           LSL = 80
       StdDev = 2.494           USL = 84

Capability indices:

        Value     2.5%   97.5%
Cp    0.26736  0.23015  0.3045
Cp_l  0.43846  0.36341  0.5135
Cp_u  0.09625  0.04028  0.1522
Cp_k  0.09625  0.02955  0.1629
Cpm   0.23785  0.20165  0.2740

Exp<LSL 9.4%     Obs<LSL 11%
Exp>USL 39%      Obs>USL 36%
>
```



Interpretation:

We learned R codings to analysis the process by Process capability analysis by Cp,Cpl, Cpu, Cpk and Cpm.

The Cp, Cpl, Cpu, Cpk and Cpm all are under unity. Therefore we conclude that the process is not in capable condition and immediate action should be taken to overcome this situation.

$Cp = \frac{USL-LSL}{6\sigma}$ $C_pL = \frac{\bar{x}-LSL}{3\sigma}$ $C_pU = \frac{LSL-\bar{x}}{3\sigma}$ $C_pk = Min(C_pL, C_pU)$ $C_pm = \frac{Cp}{\sqrt{1+\frac{(X-T)^2}{\sigma^2}}}$

PROCESS CAPABILITY ANALYSIS - II

Exercise: 51

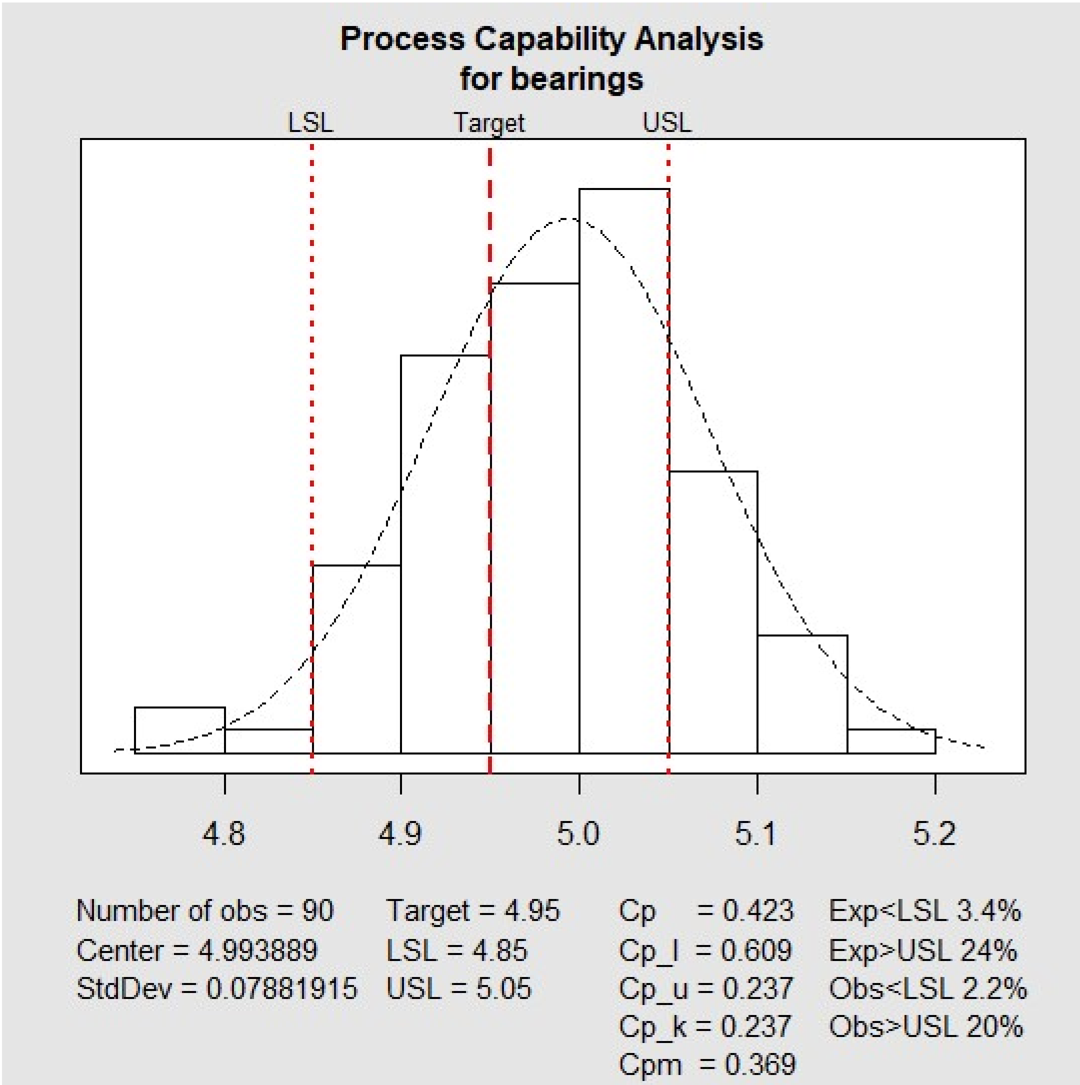
Date: 10.03.2020

Construct C_p, C_pl, C_pu, C_pk, C_pm for the following data and interpret on process control.

Hours	X1	X2	X3	X4	X5
1	5.03	5.06	4.86	4.90	4.95
2	4.97	4.94	5.09	4.78	4.88
3	5.02	4.98	4.94	4.95	4.80
4	4.97	4.93	4.90	4.92	4.96
5	5.01	4.99	4.93	5.06	5.01
6	5.00	4.95	5.10	4.85	4.91
7	4.94	4.91	5.05	5.07	4.88
8	5.00	4.98	5.05	4.96	4.97
9	4.99	5.01	4.93	5.10	4.98
10	5.03	4.98	4.92	5.01	4.93
11	5.02	4.88	5.00	4.98	5.09
12	5.09	5.01	5.13	4.89	5.02
13	4.90	4.93	4.97	4.98	5.12
14	5.04	4.96	5.15	5.04	5.02
15	5.09	4.90	5.04	5.19	5.03
16	5.10	5.01	5.04	5.05	5.02
17	4.97	5.10	5.12	4.92	5.04
18	5.01	4.99	5.06	5.04	5.12

R code and output:

```
> library(qcc)
> data<-file.choose()
> data<-read.csv(data)
> bearings<-data[,-1]
> q<-qcc(bearings,type="xbar",nsigmas=3,plot=F)
> process.capability(q,spec.limits=c(4.85,5.05))

Process Capability Analysis

Call:
process.capability(object = q, spec.limits = c(4.85, 5.05))

Number of obs = 90           Target = 4.95
       Center = 4.994           LSL = 4.85
       StdDev = 0.07882         USL = 5.05

Capability indices:

       Value    2.5%   97.5%
Cp    0.4229  0.3608  0.4849
Cp_l  0.6085  0.5138  0.7032
Cp_u  0.2373  0.1725  0.3021
Cp_k  0.2373  0.1601  0.3145
Cpm   0.3695  0.3095  0.4293

Exp<LSL 3.4%     Obs<LSL 2.2%
Exp>USL 24%      Obs>USL 20%
```



Interpretation:

We learned R coding to analysis the process by **Process Capability Analysis**.

The Cp, CpI, Cpu, CpK and Cpm all are under unity. Therefore we conclude that the process is not in capable condition and immediate action should be taken to overcome this situation.

OPERATING CHARACTERISTIC CURVE

Exercise: 52

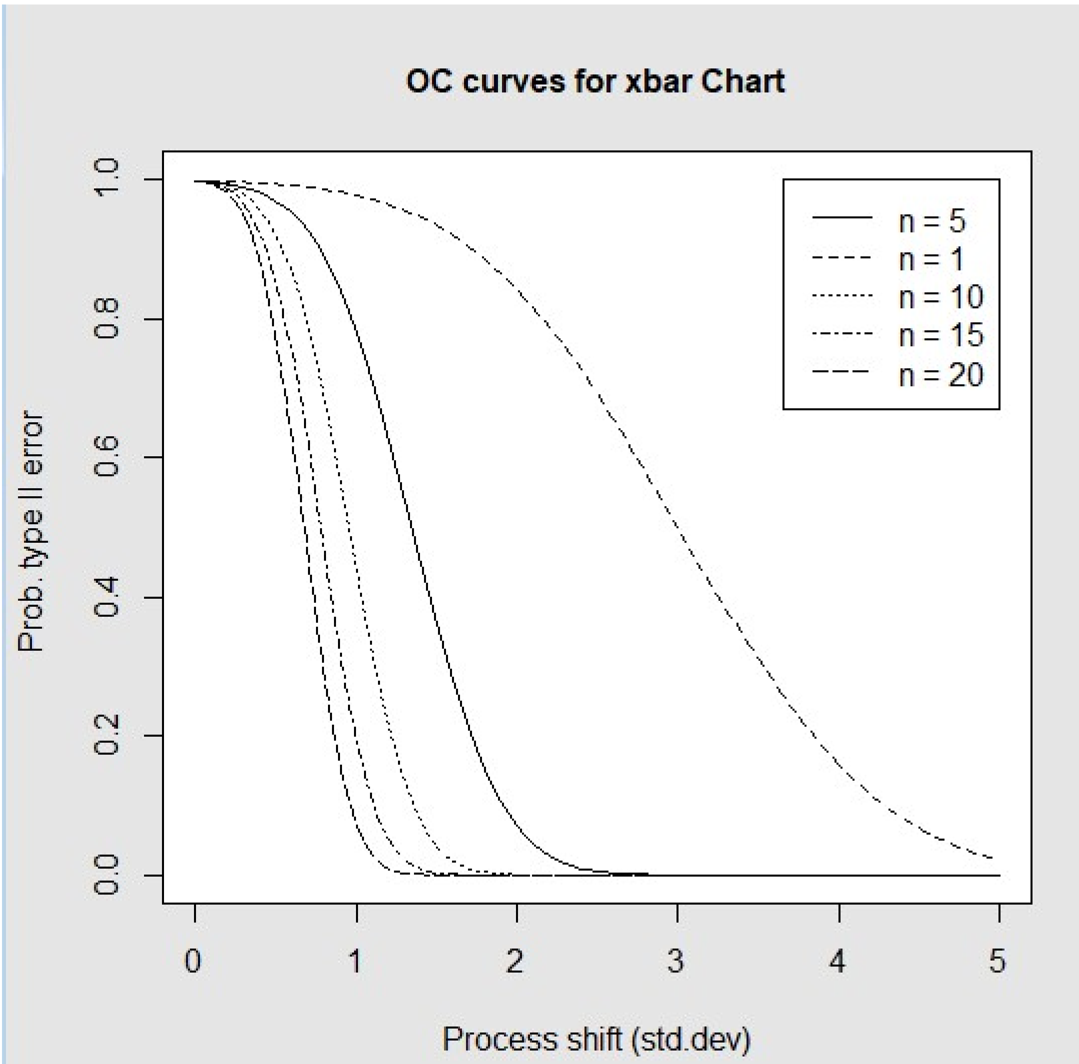
Date: 10.03.2020

Construct OC curve based on X-bar chart for sample sizes n = 1, 5, 10, 15 and 20

Sample	X1	X2	X3	X4	X5
1	83	79	81	82	83
2	83	81	85	87	81
3	85	87	83	84	86
4	80	81	83	84	83
5	83	84	85	83	84
6	88	87	89	90	88
7	80	81	82	84	81
8	79	89	88	89	89
9	78	83	85	86	93
10	88	83	82	85	82
11	78	80	78	82	81
12	81	85	85	85	84
13	77	82	84	85	87
14	81	85	85	85	84
15	85	87	82	85	89
16	83	83	77	81	80
17	85	84	84	80	82
18	82	83	80	80	83
19	75	77	84	77	78
20	85	85	86	83	80

R Code and Output:

```
> library(qcc)
> #---------------------------------OC Curve--------------------------------#
> data=read.csv(file.choose())
> piston<-data[,-1]
> beta<-oc.curves.xbar(qcc(piston,type="xbar",nsigma=3,plot=F))
> print(round(beta,digits = 4))
                sample size
shift (std.dev)    n=5    n=1   n=10   n=15   n=20
           0    0.9973 0.9973 0.9973 0.9973 0.9973
           0.05 0.9971 0.9973 0.9970 0.9968 0.9966
           0.1  0.9966 0.9972 0.9959 0.9952 0.9944
           0.15 0.9957 0.9970 0.9940 0.9920 0.9900
           0.2  0.9944 0.9968 0.9909 0.9869 0.9823
           0.25 0.9925 0.9964 0.9864 0.9789 0.9701
           0.3  0.9900 0.9960 0.9798 0.9670 0.9514
           0.35 0.9866 0.9956 0.9708 0.9500 0.9243
           0.4  0.9823 0.9950 0.9586 0.9266 0.8871
           0.45 0.9769 0.9943 0.9426 0.8957 0.8383
           0.5  0.9701 0.9936 0.9220 0.8562 0.7775
           0.55 0.9616 0.9927 0.8963 0.8078 0.7055
           0.6  0.9514 0.9916 0.8649 0.7505 0.6243
           0.65 0.9390 0.9905 0.8275 0.6853 0.5371
           0.7  0.9243 0.9892 0.7842 0.6137 0.4481
           0.75 0.9071 0.9877 0.7351 0.5379 0.3616
           0.8  0.8871 0.9860 0.6809 0.4608 0.2817
           0.85 0.8642 0.9842 0.6225 0.3851 0.2115
           0.9  0.8383 0.9821 0.5612 0.3136 0.1527
           0.95 0.8094 0.9798 0.4983 0.2485 0.1059
           1    0.7775 0.9772 0.4355 0.1913 0.0705
           1.05 0.7428 0.9744 0.3743 0.1431 0.0450
           1.1  0.7055 0.9713 0.3161 0.1038 0.0275
           1.15 0.6659 0.9678 0.2622 0.0730 0.0161
           1.2  0.6243 0.9641 0.2134 0.0497 0.0090
           1.25 0.5812 0.9599 0.1703 0.0328 0.0048
           1.3  0.5371 0.9554 0.1333 0.0209 0.0024
           1.35 0.4925 0.9505 0.1022 0.0129 0.0012
           1.4  0.4481 0.9452 0.0768 0.0077 0.0006
           1.45 0.4043 0.9394 0.0564 0.0045 0.0002
```



Interpretation:

We learnt the R coding for constructing **OC curve**.

From the OC curve, we can conclude that for sample of size providess an acceptable level of beta error for detecting a shift in process, From a sample size 20 with 5% or less than 5% defectives can be accepted with 100% probability and percent defectives above 5 can be rejected or accepted with 0% probability.